



FACULTAD DE CIENCIAS ECONÓMICAS Y ADMINISTRATIVAS

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO
PARA LA MEDICIÓN DEL PULSO CARDÍACO Y SATURACIÓN DE
OXÍGENO EN LA SANGRE.

AUTOR

Oswaldo Israel Arias Juárez

AÑO

2017



FACULTAD INGENIERIA Y CIENCIAS AGROPECUARIAS

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO PARA LA
MEDICIÓN DEL PULSO CARDÍACO Y SATURACIÓN DE OXÍGENO EN LA
SANGRE.

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniero en Redes y
Telecomunicaciones

Profesor guía

MSc. David Fernando Pozo Espín

Autor

Oswaldo Israel Arias Juárez

Año

2017

DECLARACIÓN DEL PROFESOR GUÍA

Declaro haber dirigido este trabajo a través de reuniones periódicas con la Estudiante, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

David Fernando Pozo Espín
Master Universitario en Automática y Robótica
C.I. 171734014-3

DECLARACIÓN DEL PROFESOR CORRECTOR

Declaro haber revisado este trabajo, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

Jorge Luis Rosero Beltrán
Maestría en Ciencias con especialidad en Automatización
C.I. 180361018-5

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.

Oswaldo Israel Arias Juárez
C.I. 171751562-9

AGRADECIMIENTOS

Agradezco primeramente a Dios por ser mi guía, mis padres y hermanos por ser mi apoyo constante. A la universidad de las Américas que me acogió y me guio por un buen camino de conocimientos y comprensión, al Ing. David Pozo por su paciencia y dedicación al guiarme en este trabajo. Finalmente, a mis compañeros por ser parte importante para mi aprendizaje y apoyarme en este largo camino de derrotas y éxitos.

DEDICATORIA

El presente trabajo de titulación se lo dedico a mi familia especialmente a mis padres, quienes con su sabiduría y cariño han logrado guiar por el camino del bien y han logrado levantarme en cada una de mis derrotas.

A mis hermanos los cuales serán por siempre mi compañía y confidentes.

RESUMEN

El presente trabajo de titulación tiene por objetivo diseñar e implementar un sistema de monitoreo para la medición del pulso cardíaco y saturación de oxígeno en la sangre, para su desarrollo se comenzó por realizar una recopilación, análisis y selección de la información acerca del pulso cardíaco y saturación de oxígeno de la sangre. Se realizó además una investigación de los dispositivos electrónicos que existen en la actualidad para la medición de estos parámetros, y una vez teniendo en claro los conceptos de pulso cardíaco y saturación de oxígeno en la sangre y cómo la tecnología puede ayudar a prevenir o dar seguimiento a estas enfermedades, se procedió a desarrollar un sistema de monitoreo capaz de recibir una señal proveniente de un sensor de oximetría y procesarla.

En la actualidad existen dispositivos electrónicos destinados para el control del pulso cardíaco y de la saturación de oxígeno en la sangre, pero al no contar con un aplicativo móvil para su monitoreo y una página web de gestión y verificación de los datos, se ha convertido en uno de los impedimentos para que tanto paciente, como Médico tratante, puedan tener un control adecuado frente a estas enfermedades. Como consecuencia de esta observación se determinó que una solución para esta problemática es el diseño e implementación de un sistema de monitoreo para medir la saturación de oxígeno en la sangre y pulso cardíaco.

Seguidamente se analizaron los sensores y elementos electrónicos y cómo estos interactúan en el sistema de monitoreo, lo que permitió seleccionar el sensor adecuado, que cumpla con las características requeridas para la etapa de amplificación y digitalización. Seleccionado el sensor y los elementos electrónicos, se procedió a elaborar un dispositivo electrónico capaz de detectar la variación que se produce cuando existe una baja saturación de oxígeno en la sangre y de igual manera cuando existan anomalías en la frecuencia cardíaca.

Una vez realizado el dispositivo electrónico, se procedió a desarrollar un aplicativo Android, para la comparación de los valores obtenidos por el sensor y la emisión de la alarma cuando existe alguna anomalía en estos valores, los datos recibidos, serán almacenados en una base de datos que será integrada a la Web, que podrá ser visualizada a través del internet, por el Médico tratante.

Palabras clave: Saturación de oxígeno, pulso cardíaco, sensor, Android, página web, sistema de monitoreo.

ABSTRACT

The objective of this study is design and implement a monitoring system for the measurement of the cardiac pulse and saturation of oxygen, for its development began to perform a collection, analysis and selection of information about the cardiac pulse and Oxygen saturation. An investigation was also made of the electronic devices that currently exist for the measurement of these parameters, and once the concepts of heart pulse and oxygen saturation are clear and how the technology can help prevent or follow up to these diseases, a monitoring system capable of receiving a signal from an oximetry sensor.

At present, there are electronic devices designed to control heart rate and oxygen saturation, but without a mobile application for monitoring and a web page for managing and verifying the data, it has become in one of the impediments so that both patient and treating doctor can have adequate control against these diseases. As a consequence of this observation, it was determined that a solution to this problem is the design and implementation of a monitoring system to measure oxygen saturation and heart pulse.

Next, the sensors and electronic elements were analyzed and how they interact in the monitoring system, allowing the selection of the appropriate sensor, which meets the characteristics required for the stage of amplification and digitization. After selecting the sensor and the electronic elements, an electronic device was developed to detect the variation that occurs when there is a low saturation of oxygen and also when there are abnormalities in the heart rate. Once the electronic device was made, an Android application was developed, for the comparison of the values obtained and the emission of the alarm the data received will be stored in a database That will be integrated to the Web, that can be visualized through the internet, by the treating Physician.

Key words: Oxygen saturation, cardiac pulse, sensor, Android, web page, monitoring system.

ÍNDICE

INTRODUCCIÓN	1
1. CAPÍTULO I: FUNDAMENTOS TEÓRICOS	4
1.1. Sistema de comunicación inalámbrica.....	4
1.1.1 Redes según la topología.	5
1.1.2 Redes según la cobertura.....	5
1.1.3. Principales tecnologías de las comunicaciones inalámbricas.	6
1.2. Redes de área corporal y aplicaciones.	8
1.2.1. Normalización en las BAN.	9
1.2.2. Arquitectura del sistema	9
1.2.2.1. Nodo sensor.....	10
1.2.2.2. Nodo Coordinador.....	11
1.2.3. Usos y aplicaciones.	12
1.3. Tecnología bluetooth.....	13
1.4. Conexión wifi.	14
1.4.1. Modos de conexión.....	15
1.4.2. Seguridad y fiabilidad.....	16
1.5. Saturación de oxígeno en la sangre.....	18
1.6. Pulso cardíaco.	21
2. CAPÍTULO II: DIMENSIONAMIENTO DEL SISTEMA DE MONITOREO	24
2.1. Hardware	24
2.1.1. Sensor.....	25
2.1.2. Etapa de amplificación y digitalización.....	31
2.1.3. Etapa de control.....	35
2.1.4. Etapa de comunicación.....	37
2.1.5. Fuente de alimentación.....	38
2.2. Software	39
2.2.1. Base de datos.....	40
2.2.2. Aplicativo Android.....	40

2.2.3. Software de comparación y gestión de datos.	41
2.2.4. Alojamiento web.....	43
3. CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE MONITOREO	44
3.1. Sensor Nellcor DS – 100A.....	44
3.2. Etapa de amplificación y digitalización.....	46
3.2.1. Amplificación diferencial.	47
3.2.2. Convertidor analógico – digital (ADC).....	49
3.3. Etapa de control.....	56
3.4. Etapa de comunicación.....	58
3.5. SQL SERVER motor de base de datos.....	61
3.6. Diseño de la página web.....	62
3.7. Aplicativo en Android.	87
4. CAPÍTULO IV: PRUEBAS DE FUNCIONAMIENTO.....	96
4.1. Registro y creación de Médicos.....	96
4.2. Ingreso a la página web.....	98
4.3. Registro y creación de pacientes.	100
4.4. Recuperación de contraseña.	103
4.5. Cambio de contraseña.....	104
4.6. Ingreso al aplicativo Android.....	106
4.7. Adquisición y comparación de los datos obtenidos por el sistema de monitoreo.....	107
4.8. Registro de eventos para el control continuo del paciente. .	110
5. CAPÍTULO V: ANALISIS COSTO-BENEFICIO.....	113
6. CONCLUSIONES Y RECOMENDACIONES	118
6.1. Conclusiones.....	118
6.2. Recomendaciones.....	120
REFERENCIAS	121
ANEXOS	126

ÍNDICE DE FIGURAS

Figura 1. Tipo de redes inalámbricas según su cobertura.....	6
Figura 2. Red de Área Corporal típica.....	8
Figura 3. Arquitectura de una red BAN.	10
Figura 4. Diagrama de bloques de un nodo sensor.....	11
Figura 5. Modo infraestructura.	16
Figura 6. Modo ad hoc.	16
Figura 7. Sensor desechable Nellcor	20
Figura 8. Sensor reusable Nellcor.....	21
Figura 9. Frecuencímetro cardíaco.	23
Figura 10. Diagrama de bloques del diseño del prototipo.	25
Figura 11. Sensor Nellcor.....	26
Figura 12. Funcionamiento de un sensor de oximetría	29
Figura 13. Esquema del funcionamiento del sensor Nellcor DS-100A.....	30
Figura 14. Conector DB9 hembra del sensor Nellcor.	31
Figura 15. Distribución de pines del micro controlador Atmega 164 P.....	34
Figura 16. Conexión interna de los leds del microcontrolador Atmega 164 P. .	35
Figura 17. Integrado L293D y su configuración de pines.	36
Figura 18. Módulo Bluetooth HC – 06	37
Figura 19. Esquema general del sistema	39
Figura 20. Señal obtenida a la salida del sensor Nellcor DS – 100 A.....	44
Figura 21. Estructura del registro ADMUX.	47
Figura 22. 56 muestras/4 segundos correspondientes al led rojo	52
Figura 23. 56 muestras/4 segundos correspondientes al led infrarrojo	52
Figura 24. Pulsos correspondientes al led rojo e infrarrojo	53
Figura 25. Encendido del led rojo e infrarrojo.....	¡Error! Marcador no definido.
Figura 26. Procesamiento de muestras del led rojo	55
Figura 27. Procesamiento de muestras del led infrarrojo	56
Figura 28. Etapa de control del sistema de monitoreo de control.....	57
Figura 29. Diagrama circuital del prototipo del sistema de monitoreo y control del pulso cardíaco y saturación de oxígeno en la sangre.....	60

Figura 30. Diagrama circuital de la fuente de alimentación.....	60
Figura 31. Diseño e la base de datos “BDSISTEMA”	61
Figura 32. Formulario 1	62
Figura 33. mensajes de error del formulario 1	63
Figura 34. Registro y creación de usuario.....	63
Figura 35. Mensajes de error del formulario 1	64
Figura 36. Mensaje de creación de usuario	65
Figura 37. Formato de mensaje enviado al mail personal.....	65
Figura 38. Diagrama de flujo del formulario 1.....	66
Figura 39. Diagrama de flujo del registro y creación de usuario.....	68
Figura 40. Formulario 2 si se escoge en sexo la opción Mujer.....	68
Figura 41. Formulario 2 si se escoge en sexo la opción Hombre.....	69
Figura 42. Formulario 2 si se escoge en el sexo “Sin especificar”.	69
Figura 43. Diagrama de flujo del formulario 2.....	70
Figura 44. Presentación del formulario 3.....	71
Figura 45. Mensaje de error si el paciente ya existe	71
Figura 46. Mensaje de error para espacios en blanco o correo mal escrito	72
Figura 47. Mensaje de creación de paciente.....	72
Figura 48. Formato de mensaje enviado al mail personal del paciente.....	72
Figura 49. Diagrama de flujo del formulario 3.....	73
Figura 50. Presentación del formulario 4.....	74
Figura 51. Diagrama de flujo del formulario 4.....	75
Figura 52. Diseño de la página maestra.....	76
Figura 53. Diseño de la opción recordar contraseña	77
Figura 54. Mensaje de error si los datos no están registrados	77
Figura 55. Mensaje de error si los dos espacios están en blanco	77
Figura 56. Mensaje de aviso que la contraseña fue recuperada	78
Figura 57. Formato de mensaje enviado al mail personal	78
Figura 58. Diagrama de flujo de la opción recuperar contraseña	79
Figura 59. Diseño de la opción cambiar contraseña	80
Figura 60. Mensaje de error si los dos espacios están en blanco	80
Figura 61. Mensaje de error si las contraseñas no coinciden.....	81
Figura 62. Mensaje de error si la contraseña ya existe	81

Figura 63. Mensaje de aviso que la contraseña ha sido reemplazada.....	81
Figura 64. Formato de mensaje enviado al mail personal.....	82
Figura 65. Diagrama de flujo de la opción cambiar contraseña.....	83
Figura 66. Funcionamiento del Webservice.....	84
Figura 67. Diagrama de flujo para el envío de correos.....	86
Figura 68. Diseño del layout 1.....	87
Figura 69. Mensaje de error si el campo para inicio de sesión esta en blanco	88
Figura 70. Mensaje de error a una clave de inicio de sesión errónea.....	88
Figura 71. Diagrama de flujo correspondiente al layout 1.....	89
Figura 72. Solicitud de permiso para encender Bluetooth.....	91
Figura 73. Sincronización del aplicativo Android con el prototipo.....	91
Figura 74. Transferencia de datos.....	92
Figura 75. Mensaje de alerta.....	92
Figura 76. AndroidManifest.xml y su configuración básica para el funcionamiento del prototipo.....	93
Figura 77. Diagrama de flujo del programa.....	95
Figura 78. Tabla Doctores de la base de datos BDSISTEMA sin registros.....	96
Figura 80. Nuevo registro en la tabla Doctores de la base de datos BDSISTEMA.....	97
Figura 81. Mensaje personal de correo electrónico con los datos del nuevo registró.....	97
Figura 82. Ingreso al software de comparación y gestión de datos.....	99
Figura 83. Tabla Ingreso_Paciente de la base de datos BDSISTEMA sin registros.....	100
Figura 84. Registro y creación de un paciente.....	101
Figura 85. Proceso de creación de nuevo paciente por médico tratante.....	102
Figura 86. Tabla Doctores de la base de datos BDSISTEMA.....	103
Figura 87. Solicitud de envío de contraseña al correo electrónico.....	103
Figura 88. Mensaje recibido con usuario y contraseña del médico tratante...	104
Figura 89. Proceso de cambio de contraseña.....	105
Figura 90. Mensaje personal de correo electrónico con los nuevos datos de usuario y contraseña.....	106
Figura 91. Proceso de ingreso al aplicativo Android.....	107
Figura 92. Valores del pulso cardíaco y saturación de oxígeno en la sangre	108

Figura 93. Tabla Control_Paciente de la base de datos BDSISTEMA sin registros.....	111
Figura 94. Activación de la alarma en el aplicativo Android	111
Figura 95. Nuevo registró en la tabla Control_Paciente de la base de datos BDSISTEMA	112

ÍNDICE DE TABLAS

Tabla 1. Comparación de las comunicaciones inalámbricas.....	7
Tabla 2. Comparación de los estándares wifi.....	15
Tabla 3. Actuación según % de saturación de oxígeno.....	19
Tabla 4. Pulso cardíaco en niños.....	22
Tabla 5. Pulso cardíaco en adultos.....	22
Tabla 6. Característica de los sensores.....	26
Tabla 7. Características del sensor Nellcor DS – 100A.....	27
Tabla 8. Descripción de los pines del sensor.....	31
Tabla 9. Características del micro controlador Atmega 164P.....	33
Tabla 10. Descripción de los pines del micro controlador Atmega 164 P.....	34
Tabla 11. Descripción de los pines del integrado L293D.....	36
Tabla 12. Características batería Samsung B500AE.....	38
Tabla 13. Características del circuito integrado regulador L7805CV.....	38
Tabla 14. Tabla comparativa de los lenguajes de programación.....	42
Tabla 15. Cantidad de pulsos que el corazón puede dar por minuto.....	45
Tabla 16. Descripción de los Bits del registro ADMUX.....	47
Tabla 17. Combinación de los Bits MUX4:0 del registro ADMUX.....	48
Tabla 18. Combinación de los Bits en el registro ADMUX.....	48
Tabla 19. Combinación de los Bits REFS1 y REFS0 del registro ADMUX.....	49
Tabla 20. Control de encendido de los leds.....	58
Tabla 21. Pruebas de funcionamiento en primera persona.....	108
Tabla 22. Pruebas de funcionamiento en segunda persona.....	109
Tabla 23. Pruebas de funcionamiento en tercera persona.....	109
Tabla 24. Pruebas de funcionamiento en cuarta persona.....	110
Tabla 25. Pruebas de funcionamiento en quinta persona.....	110
Tabla 26. Características del oxímetro Iincare Bluetooth Oximeter.....	113
Tabla 27. Características del oxímetro wirelles bluetooth apple gima.....	114
Tabla 28. Tabla de costos del sistema de monitoreo y control.....	114
Tabla 29. Características del sistema de monitoreo y control.....	115
Tabla 30. Tabla comparativa de los 3 dispositivos de medición.....	116

INTRODUCCIÓN

El sistema de monitoreo para la medición del pulso cardíaco y saturación de oxígeno en la sangre proporciona datos informativos sobre la cantidad de oxígeno presente en la sangre, y del pulso cardíaco de una persona mediante procedimientos no invasivos.

El sistema de monitoreo está constituido por cuatro etapas, distribuidas de la siguiente manera:

- **Primera:** Está formada por un dispositivo electrónico que contiene un sensor, el mismo que irá adherido al dedo índice de la mano; el dispositivo con ayuda del sensor medirá los valores producidos por la saturación de oxígeno en la sangre y el pulso cardíaco.
- **Segunda:** Los valores procesados en la primera etapa, serán enviados a través de tecnología Bluetooth a un dispositivo móvil, para realizar una comparación con los valores de saturación de oxígeno normal mediante un programa desarrollado en Android, y verificar si luego de esa comparación es necesario emitir al dispositivo móvil una alarma.
- **Tercera:** El diseño, creación e integración de una base de datos a la Web, conforman la tercera etapa, esta base de datos almacena los valores receptados por el dispositivo móvil, el registro de los médicos, y el ingreso y creación de los pacientes.
- **Cuarta:** Una vez realizado todo el proceso anterior; como última etapa del sistema de monitoreo se tiene el diseño de una página web, para que el médico tratante pueda acceder a la base de datos y verificar la información de sus pacientes.

Para la realización del sistema de monitoreo fue necesario conocer los conceptos básicos de saturación de oxígeno en la sangre y pulso cardíaco.

PLANTEAMIENTO DEL PROBLEMA

Los médicos necesitan tener un control continuo del ritmo cardíaco y saturación de oxígeno de sus pacientes, pero es difícil que ellos estén presentes en cada actividad que sus pacientes realizan para la obtención de los datos. Con el diseño del sistema de monitoreo se soluciona este inconveniente ya que el paciente puede llevar consigo el dispositivo de monitoreo, con el fin de obtener los datos del sensor, correspondiente a los pulsos del corazón y saturación de oxígeno, y a la vez, si existe alguna anomalía en los datos obtenidos estos serán automáticamente almacenados en una base de datos para que el médico tratante pueda tener acceso a ellos a través del internet, sin necesidad de que paciente y médico se vean personalmente.

El sistema de monitoreo que se propone en el presente trabajo de titulación cumple con funciones similares a los dispositivos existentes en el mercado Ecuatoriano destinados para el control de saturación de oxígeno en la sangre y pulso cardíaco, pero adicional tiene algunas funciones agregadas, como son: aplicativo móvil para Android, alarma de notificación, base de datos y diseño de página web.

OBJETIVOS

OBJETIVO GENERAL

Diseñar e implementar un sistema de monitoreo para la medición del pulso cardíaco y saturación de oxígeno en la sangre, mediante un estudio y análisis comparativo de los dispositivos, elementos electrónicos y tecnologías.

OBJETIVOS ESPECÍFICOS

- Estudiar y analizar los requerimientos del sistema para diseñar el prototipo.
- Dimensionar el sistema de monitoreo.
- Implementar el prototipo del sistema de monitoreo.
- Realizar pruebas de funcionamiento del sistema.
- Realizar un análisis de costo-beneficio del sistema de monitoreo para la medición del pulso cardíaco y saturación de oxígeno en la sangre.

1. CAPÍTULO I: FUNDAMENTOS TEÓRICOS

Hoy en día la humanidad cuenta con los aportes que la tecnología le ha brindado en las diferentes disciplinas académicas, todos los avances tecnológicos que están presentes en la vida cotidiana son de vital importancia en el desarrollo y evolución de la sociedad.

La medicina en general se ha favorecido de los aportes que le ha brindado el desarrollo tecnológico como, por ejemplo, las Redes de Área Personal (PAN. Personal Area Network) y las redes de Área Corporal (BAN. Body Area Network), gracias a estas tecnologías se ha logrado dar grandes cambios en la medicina, se ha permitido conocer el origen y causa de muchas enfermedades, así como su posible prevención y tratamiento. (Betancur, 2014).

1.1. Sistema de comunicación inalámbrica.

Los sistemas de comunicación inalámbrica se definen como sistemas de comunicación en donde, emisor y receptor se comunican mediante la modulación de ondas electromagnéticas a través del espacio, es decir tanto emisor como receptor no necesitan estar unidos de manera física para establecer una comunicación. (Prieto, 2013).

Se pueden encontrar diferentes clasificaciones de sistemas de comunicación inalámbrica, pero para interés del presente trabajo de titulación, se las ha clasificado de acuerdo a su topología y cobertura.

1.1.1 Redes según la topología.

- **Red de difusión.**

Son aquellas redes que comparten un canal de comunicación para varias máquinas conectadas a la red.

- **Red punto – punto.**

Son aquellas redes, en las cuales se puede tener varias conexiones, pero cada canal de datos es usado únicamente para comunicar 2 máquinas.

1.1.2 Redes según la cobertura.

- **BAN (Body Area Network).**

Es un tipo de red de comunicación inalámbrica, entre dispositivos de baja potencia utilizados en el cuerpo humano.

- **WPAN (Wireless Personal Area Network).**

Es una red de cobertura personal y su alcance se limita a un área muy reducida, como, por ejemplo: una habitación u oficina.

- **WLAN (Wireless Local Area Network).**

Estas redes tienen cobertura limitada, es decir abarca pocos metros, es ideal, para oficinas, hogares, edificios, e incluso se las puede utilizar en un campus universitario.

- **WMAN (Wireless Metropolitan Area Network).**

Estas redes tienen mayor cobertura, y se puede cubrir sin ningún problema grandes áreas como una ciudad, ya que su cobertura alcanza varios kilómetros.

- **WWAN (Wireless Wide Area Network).**

Son redes destinadas a cubrir grandes extensiones, tales como un país e incluso el planeta tierra. (Prieto, 2013)

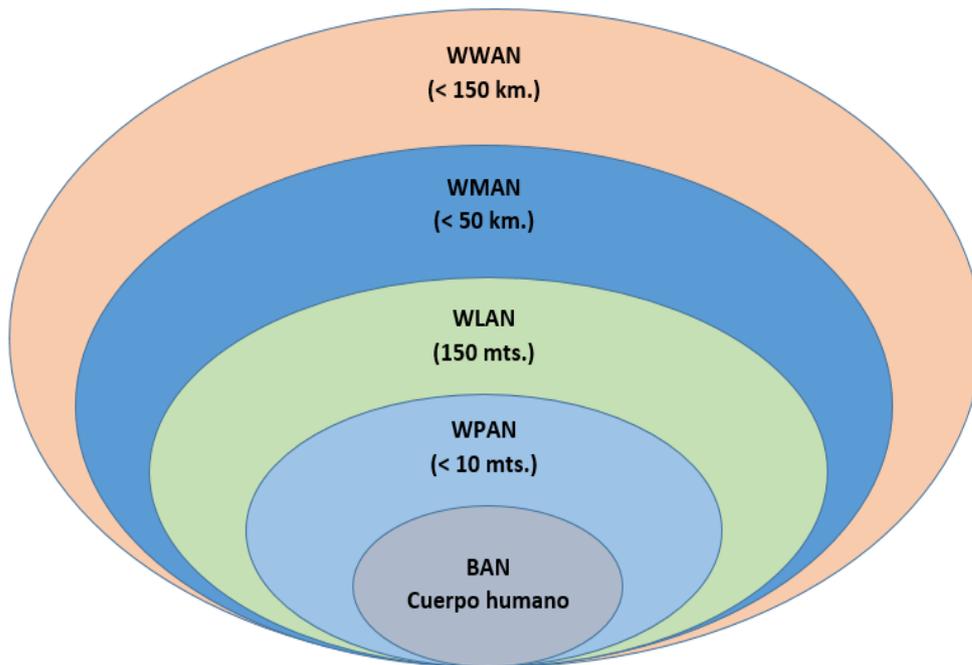


Figura 1. Tipo de redes inalámbricas según su cobertura.

1.1.3. Principales tecnologías de las comunicaciones inalámbricas.

Hay gran variedad de tecnologías destinadas para las comunicaciones inalámbricas, pero depende de la aplicación para su utilización; cada tecnología tiene sus respectivas ventajas y desventajas. Para el presente proyecto se listarán las más importantes.

- **802.11B:** Es la tecnología usada con mayor frecuencia para redes inalámbricas de ordenadores y para compartir acceso a internet, amplía la tasa de transferencia hasta los 11 Mbps.
- **Infrarrojo:** El infrarrojo se lo encuentra en los dispositivos portátiles como computadores personales, impresoras y notebooks, utiliza altas frecuencias para el transporte de los datos dentro de una distancia de 5-10 m.

- **Bluetooth:** Transmisión de datos por radiofrecuencia, Bluetooth permite la transmisión de datos y voz entre diferentes dispositivos que se encuentren a una corta distancia, por lo general no mayor a 10 m.
- **Wifi:** Mecanismo para la conexión de dispositivos a la red, sin necesidad de usar un medio físico para la conexión a internet.
- **IEE 802.15.6.:** Estándar destinado para las redes BAN.
- **Microondas:** Se definen como ondas electromagnéticas utilizadas para las redes inalámbricas, por lo general se las usa en radares, telefonía, televisión y comunicación satelital
- **UMTS:** La tecnología UMTS es una de las principales tecnologías usadas por la comunicación móvil de tercera generación o 3G.

Tabla 1.

Comparación de las comunicaciones inalámbricas

	Velocidad de tx.	Distancia	Banda de frecuencia	Aplicaciones
802.11 B	11 Mbps	< 91 m.	2.4 Ghz.	Aplicaciones militares
Infrarrojo	4 Mbps	(5 – 10) m.	-	Electrodomésticos
Bluetooth	32 Mbps	< 10 m.	2.4 Ghz.	Teclados, micrófonos.
Wifi	1.3 Gbps	Hasta 150 m.	(2.4 y 5) Ghz.	Internet
IEE 802.15.6.	10 Mbps	(75 – 100) m.	-	Redes BAN
Microondas	750 Mbps	< 150 Km.	-	Internet
Umts	2 Mbps	-	(1.885-2.025) MHz y (2.110-2.200) MHz	Telecomunicaciones móviles.

Tomado de (SciELO, s.f.)

1.2. Redes de área corporal y aplicaciones.

Las redes de área corporal (BAN), se definen como sistemas de comunicaciones a pequeña escala, es decir que la transmisión de los datos se lo lleva a cabo, alrededor o internamente en el cuerpo humano. (Betancurt, 2011).

Una red BAN típica como se puede apreciar en la Figura 2, está formada por dispositivos implantados o adheridos al cuerpo humano.

Los dispositivos representados en la Figura 2 controlan los parámetros vitales de un ser vivo, estos dispositivos se comunican entre sí mediante un sistema de comunicación inalámbrica, transmiten los datos del cuerpo a una estación base por ejemplo a un dispositivo móvil, y a su vez dichos datos pueden ser enviados a un centro de salud en tiempo real.

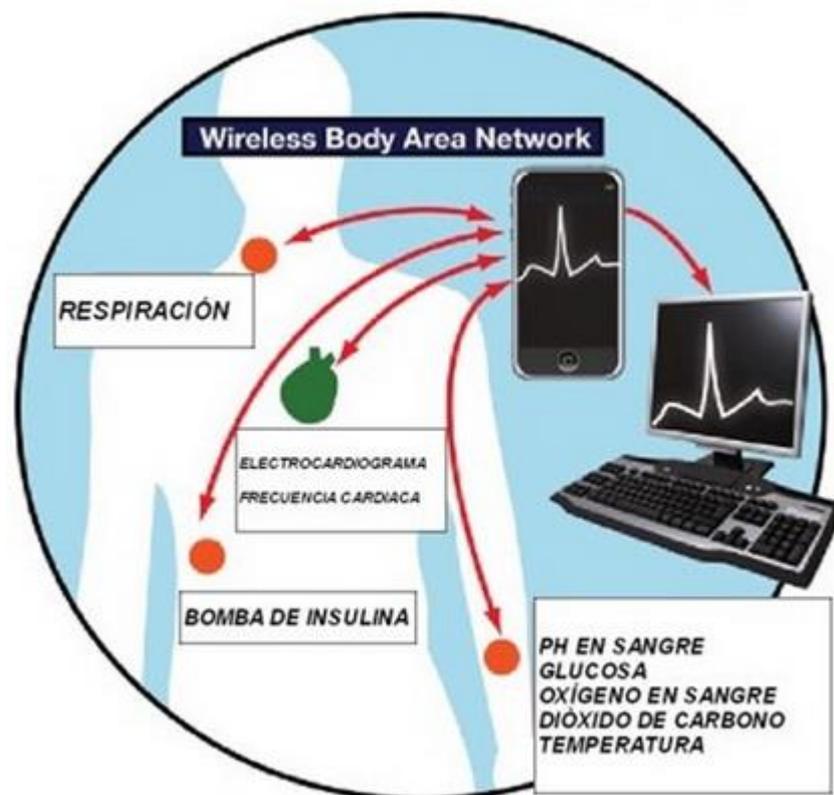


Figura 2. Red de Área Corporal típica.

Tomado de (Cibermedico, s.f.)

1.2.1. Normalización en las BAN.

La IEEE (Instituto de Ingeniería Eléctrica y Electrónica) aprobó el estándar IEEE 802.15.6 para el desarrollo y estudio de las redes BAN que es un estándar de bajo consumo de potencia y actúa dentro, sobre o alrededor del cuerpo humano.

Entre las principales características del IEEE 802.15.6 se pueden destacar las siguientes:

- Poseen un alcance hasta 75 o 100 m.
- Soporta calidad de servicio (QoS).
- Velocidades de transmisión hasta 10Mbps.
- Se adapta fácilmente a los movimientos de los usuarios.

“Los estándares Bluetooth (IEEE 802.15.1), ZigBee (IEEE 802.15.4), las propuestas de Ultra Wide Band en (IEEE 802.15.4a), el estándar europeo ECMA 368 - ISO/IEC 26907 y las regulaciones de la FCC del 2002 para las redes de Ultra Wide Band, no pertenecen a la estandarización para redes de área corporal, pero al ser aplicado dentro de las redes de área personal, cumplen con los requisitos suficientes para ser tomados en cuenta”. (Betancur, 2014).

1.2.2. Arquitectura del sistema

Una red BAN, está constituida por un nodo coordinador y uno o varios nodos sensores como se muestra en la Figura 3.

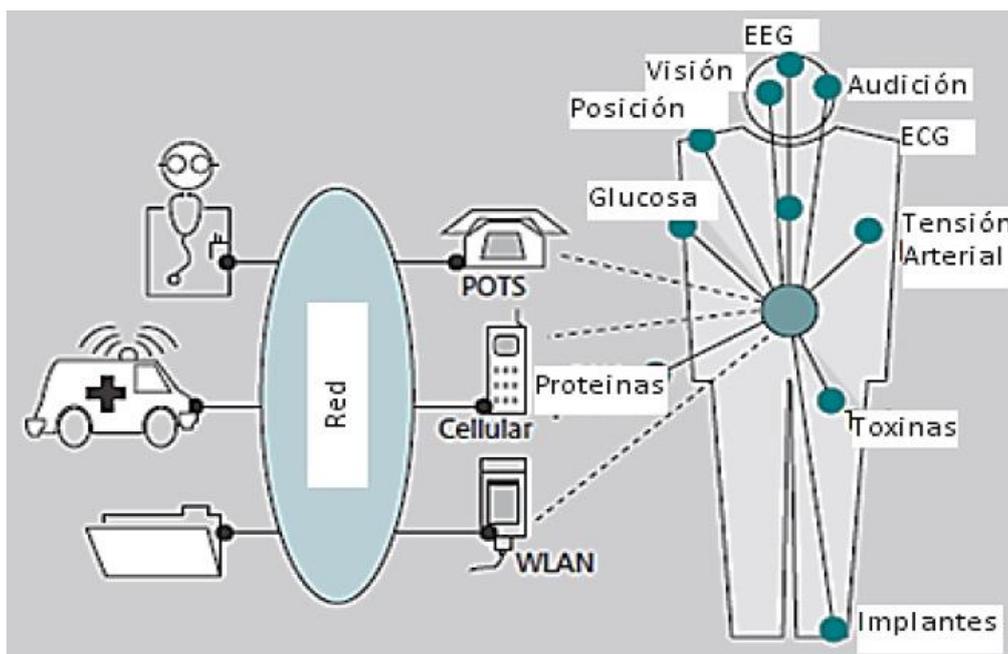


Figura 3. Arquitectura de una red BAN.

Tomado de (Tecno revolución, 2014)

La comunicación entre el nodo sensor y el nodo coordinador se realiza de forma directa, pero si no existe una línea de vista entre estas 2 partes se coloca un nodo repetidor, para que funcione como puente de comunicación entre el nodo coordinador y el nodo sensor.

1.2.2.1. Nodo sensor.

El nodo sensor va adherido en una parte específica del cuerpo humano como puede ser dedo, oído, entre otras, estos nodos recopilan y procesan información, para luego almacenarla localmente y enviarla al nodo coordinador. Un nodo sensor está compuesto por los siguientes componentes:

- Módulo sensor.
- Módulo de radiofrecuencia.
- Módulo de memoria.
- Módulo del microprocesador.

En la Figura 4 se muestra un diagrama de bloques del nodo sensor.

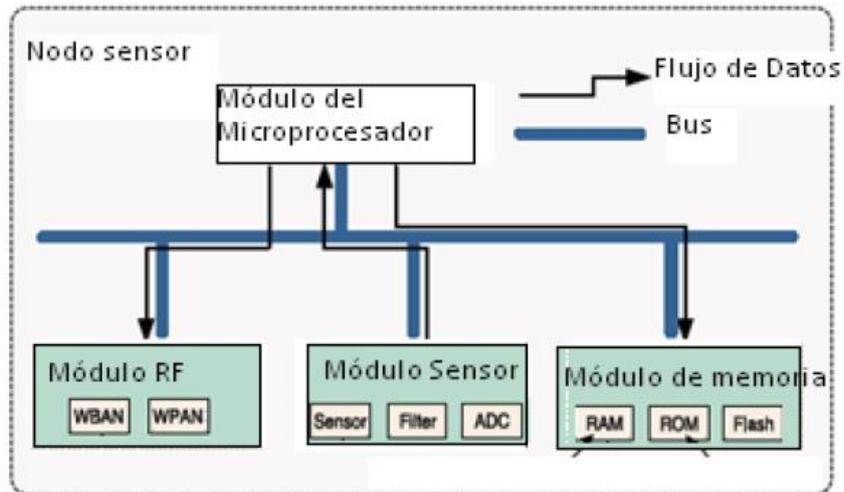


Figura 4. Diagrama de bloques de un nodo sensor.

Tomado de (Alvarado, 2012)

El módulo sensor contiene específicamente un sensor, un filtro y un convertidor analógico – digital; el sensor transforma la señal fisiológica a una señal eléctrica, luego esta señal eléctrica pasa por un filtro pasa banda y se digitaliza a través del convertidor analógico – digital. La señal digitalizada se almacena en memoria y con la ayuda del módulo de radiofrecuencia la señal es enviada hacia el nodo coordinador.

1.2.2.2. Nodo Coordinador.

El nodo coordinador realiza las siguientes funciones:

- Inicialización, configuración y sincronización de los nodos sensores.
- Recopilación de la información de los nodos sensores.
- Comunicación con los servidores remotos del proveedor de atención médica utilizando los servicios de Internet o algún otro tipo de enlace.

Toda la información y los datos obtenidos de los sensores son procesados por el nodo coordinador que puede ser un teléfono celular o una computadora personal.

“Los nodos sensores envían los datos a los nodos coordinadores, a través de una red inalámbrica de corto alcance (Bluetooth, ZigBee o UWB), y a su vez el nodo coordinador transmite los datos al servidor remoto mediante una red de área local inalámbrica (WLAN) o algún otro tipo de enlace”. (Alvarado y Juárez, 2012).

1.2.3. Usos y aplicaciones.

El objetivo de las redes BAN es proporcionar confidencialidad, vigilancia y control de la salud, entre sus principales usos y aplicaciones se pueden mencionar las siguientes:

- Aplicaciones médicas.
- Electrónica de consumo.
- Entretenimiento.

Se tiene varias aplicaciones donde son implementadas las redes BAN, a continuación, se presentan las principales y las cuales fueron seleccionadas de acuerdo a su campo de aplicación y operación.

- **Monitorización sanitaria remota:** Debido a la gran demanda de personas que tienen sobrepeso y enfermedades crónicas, las cuales les imposibilita trasladarse al hospital para un chequeo médico, se ha implementado la vigilancia convencional a la monitorización ambulatoria de pacientes para de esta manera poder mantener un control adecuado y pertinente cuando el paciente se encuentra fuera del hospital.

- **Entrenamiento deportivo:** Actualmente existen dispositivos los cuales pueden ser adheridos en partes específicas del cuerpo humano, con el objetivo de identificar y corregir diferentes posturas y movimientos específicos que realiza una persona, uno de estos dispositivos es el acelerómetro. Gracias a estos dispositivos los deportistas han podido mejorar su rendimiento, evitar y corregir lesiones provocadas por malas posturas.
- **Operaciones militares:** Las redes BAN implementadas en operaciones militares han colaborado con las siguientes funciones:
 - Reducir la probabilidad de algún daño físico al personal provocado por las condiciones ambientales.
 - Mejorar la atención médica en caso de existir alguna lesión.

1.3. Tecnología bluetooth.

Bluetooth es una tecnología usada en las comunicaciones inalámbricas su característica principal es generar comunicación entre dispositivos con el objetivo de poder transmitir voz y datos a través de un enlace de radiofrecuencia con un alcance no mayor a 10 m.

Las primeras aplicaciones desarrolladas con Bluetooth fueron enfocadas para personas que viajan constantemente y necesitaban transferencia de datos sin tener que estar conectados a Internet. (Millan, 2012).

En la actualidad se utiliza Bluetooth principalmente en teléfonos, impresoras, módems y auriculares posibilitando de esta manera aplicaciones como:

- Conexión entre celulares.
- Comunicación entre computadora y periféricos de entrada y salida sin necesidad de tener un cableado.
- Transferencia de documentos, archivos, presentaciones entre usuarios sin necesidad de cables.
- Atender llamadas telefónicas con un auricular o manos libres sin necesidad de cable.
- Uso del teléfono móvil como control remoto.

1.4. Conexión wifi.

Wifi es una tecnología de comunicación inalámbrica que permite la conexión a internet de dispositivos sin necesidad de tener un cableado de por medio y comprende un grupo de estándares basada en las especificaciones IEEE 802.11.

Los estándares Wifi que se tiene hoy en día fueron creados por la organización Wifi Alliance, y cada uno de ellos está basado en el estándar IEEE 802.11.

- **IEEE 802.11:** Fue el primer estándar creado en 1997, actualmente está en desuso y no es compatible con los dispositivos actuales, su máxima conexión de velocidad es 2 Mbps.
- **IEEE 802.11a:** Su velocidad máxima de conexión es 54Mbps, opera en la frecuencia de 5 Ghz, pero su alcance es limitado debido a que en la frecuencia de 5 Ghz, la señal es bloqueada fácilmente por cualquier objeto.
- **IEEE 802.11b:** Su velocidad máxima de conexión es de 11Mbps, opera en la frecuencia de 2,4 Ghz.

- **IEEE 802.11g:** Su velocidad máxima de operación es de 54 Mbps, opera en la frecuencia de 2,4 Ghz, debido a que su velocidad es la adecuada para varias aplicaciones, este estándar sigue siendo utilizado hoy en día.
- **IEEE 802.11n:** Es un estándar creado en el 2009, trabaja en las frecuencias de 2,4 y 5 Ghz, con una velocidad máxima de 600 Mbps.

En la tabla 2 se presenta un cuadro comparativo de los estándares Wifi existentes.

Tabla 2.

Comparación de los estándares wifi.

	802.11	802.11a	802.11b	802.11g	802.11n
Año de creación	1997	1999	1999	2003	2009
Frecuencia	-	5 Ghz	2,4 Ghz	2,4 GHz	2,4 GHz y 5 GHz
Velocidad	2 Mbps	54 Mbps	11 Mbps	50 Mbps	600 Mbps
Compatibilidad	-	802.11 a/n	802.11 b/g/n	802.11 b/g/n	802.11 a/b/g/n
Comentarios	Obsoleto	Limitado			Mejor alcance

Tomado de. (Cemeblog. s.f.)

1.4.1. Modos de conexión.

El estándar 802.11 define dos modos de conexión:

- **Modo infraestructura.**

En este modo de conexión, cada estación base se conecta a un punto de acceso por medio de un enlace inalámbrico, como se puede observar en la Figura 5.

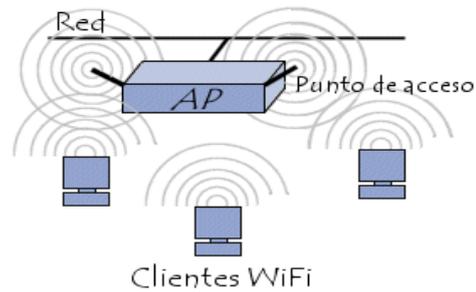


Figura 5. Modo infraestructura.

Tomado de (CCM, s.f.)

- **Modo ad hoc.**

En el modo ad hoc los dispositivos o equipos se conectan entre sí, formando de esta manera una red punto a punto, donde cada equipo funciona como cliente y punto de acceso, como se puede observar en la Figura 6.

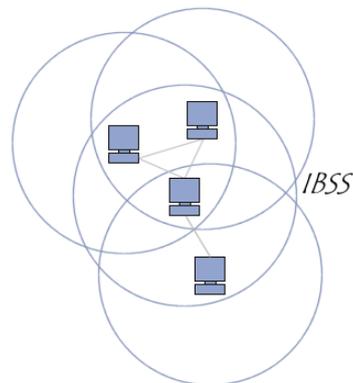


Figura 6. Modo ad hoc.

Tomado de (CCM. s.f.)

1.4.2. Seguridad y fiabilidad.

Debido a la gran demanda de usuarios que requieren la conexión a internet de diversas aplicaciones o servicios, el espectro electromagnético se ha ido saturando, convirtiéndose en uno de los principales problemas para la tecnología Wifi, ya que afecta severamente a las conexiones con una distancia mayor a 100m.

Gran cantidad de redes son instaladas con dispositivos wifi cuya configuración es poco segura, como, por ejemplo; un router, al ser instalado sin ningún tipo de seguridad es fácilmente saber la clave de acceso a la red y tener un control del dispositivo.

Para evitar este inconveniente es necesario seguir los siguientes consejos:

- Cambios continuos de la contraseña de acceso.
- Modificar el SSID predeterminado.
- Indicar los dispositivos autorizados para conectarse a la red.
- Utilización de cifrado: WPA2.
- Mediante la MAC address, filtrar los dispositivos.

La utilización de protocolos de cifrado de datos son alternativas para la seguridad de las redes wifi, debido a que codifican la información transmitida protegiendo de esta manera la confidencialidad. Entre los principales protocolos de cifrado se tienen los siguientes:

- **WEP:** WEP logra que solo el destinatario puede tener acceso a los datos, realizando un cifrado de los mismos, es decir codifica los datos mediante una clave. La desventaja es que cualquier cracker puede conseguir la clave de codificación de los datos.
- **WPA:** La clave de acceso a la codificación de los datos se inserta como dígitos alfanuméricos.
- **IPSEC:** Permite autenticación e identificación de usuarios.
- **Filtrado de MAC:** Permite el acceso a la red solamente a los equipos autorizados.
- **Ocultación de punto de acceso:** Se oculta de manera virtual el router.
- **WPA2:** Es una mejora y sucesor del WPA, su desventaja es que necesita hardware y software compatibles con los dispositivos debido a que los antiguos no lo son.

“La seguridad de una red wifi puede ser puesta a prueba mediante una auditoria de wifi. Sin embargo, no existe ninguna alternativa totalmente fiable, ya que todas ellas son susceptibles de ser vulneradas”². (Martínez, 2013).

Existe un sin número de escenarios donde se puede aplicar y usar la tecnología wifi, entre los cuales podemos mencionar los siguientes:

- Escenario residencial.
- Redes corporativas.
- Usos industriales.
- Acceso público al internet.

1.5. Saturación de oxígeno en la sangre.

La saturación de oxígeno en la sangre se define como la cantidad de oxígeno presente en el torrente sanguíneo, es decir, cuando la sangre se bombea desde el corazón hacia el resto del cuerpo, hace su viaje por los pulmones y es aquí donde las moléculas de oxígeno se unen a las células rojas de la sangre conocidas como eritrocitos, la cantidad de eritrocitos que están saturados con oxígeno se define como saturación de oxígeno en la sangre. La saturación de oxígeno en la sangre saludable y normal esta entre un 95% y un 100%, pero las personas que poseen un porcentaje bajo en eritrocitos (glóbulos rojos) se ven obligados a usar oxígeno suplementario.

En la tabla 3 se muestran los valores de saturación de oxígeno en la sangre y las indicaciones de su tratamiento.

Tabla 3.

Actuación según % de saturación de oxígeno

Actuación según % de Saturación	
% Saturación	Actuación
> 95 %	No actuación inmediata.
95 – 90 %	Tratamiento inmediato y monitorización de la respuesta al mismo, según esta, valorar derivación al hospital. Los pacientes con enfermedad respiratoria crónica toleran bien saturaciones en tornos a estos valores.
< 90 %	Enfermo grave. Hipoxia severa.
< 80 %	Valorar intubación y ventilación mecánica.
En niños con < 92 %: Remitir al hospital, aunque presenten mejoría con maniobras iniciales, por ser más incierta su respuesta al tratamiento.	

Tomado de (Noguerol, Gonzáles, 2012)

Existen varias maneras en las que se puede diagnosticar la hipoxemia (disminución anormal del oxígeno en la sangre) por ejemplo:

- **Prueba de gases con sangre:** Se toma una muestra de sangre y se la somete a análisis para determinar la concentración de oxígeno en la sangre.
- **Pulsioxímetros:** Dispositivos que miden la concentración de oxígeno en la sangre.
- **Tomografía computarizada y resonancia magnética:** Previenen enfermedades cardiacas que pueden causar una hipoxemia.

“Los períodos prolongados de niveles bajos de oxígeno en la sangre pueden ser muy graves, llegando a causar enfermedades peligrosas para el ser humano, entre las cuales se pueden destacar las siguientes:”³. (Madrid, 2014).

- Falta de aliento.
- Poca energía.
- Enfermedades mentales.
- Edema pulmonar a grandes alturas.
- Complicaciones pulmonares y cardíacas.

Métodos y dispositivos actuales de medición.

El oxímetro de pulso es el dispositivo que se utiliza para la medición y monitorización de oxígeno en algunas arterias de hemoglobina, los oxímetros de pulso tienen en su interior un sensor, este sensor está disponible para poder ser adherido en cualquier parte del cuerpo, por ejemplo, los dedos, oído, nariz y frente que son las partes más comunes para poder realizar la medición del SpO₂ (saturación de oxígeno en la sangre), estos sensores se clasifican en sensores desechables y reusables, cuya representación física se observa en las Figuras 7 y 8 respectivamente.



Figura 7. Sensor desechable Nellcor.

Tomado de (Cables y sensores, s.f.)

En la actualidad los oxímetros más usados para la medición de la saturación de oxígeno en la sangre son los siguientes:



Figura 8. Sensor reusable Nellcor.

Tomado de (Cables y sensores, s.f.)

- **Oxímetro de mano:** Llamado también como oxímetro portátil, se debe asegurar, que la batería este suficientemente cargada para el encendido de esta manera se asegura que el oxímetro no se apagará en el proceso de medición.
- **Oxímetro estático:** Para su funcionamiento y operación siempre debe estar conectado a una toma corriente ya que uno de los requisitos para su funcionamiento es que debe ser utilizado con energía eléctrica.

1.6. Pulso cardíaco.

El pulso cardíaco de una persona se define como la pulsación que provoca la expansión de sus arterias, debido a la circulación de la sangre bombeada por el corazón. Existen ciertos valores mediante los cuales se puede saber si el ritmo cardíaco en una persona es el correcto, dichos valores están representados en las tablas 4 y 5. Cabe aclarar que los datos de las tablas 4 y 5 fueron tomados de personas en estado de reposo.

Tabla 4.

Pulso cardíaco en niños

Pulso cardíaco en niños	
Edad	Frecuencia Cardíaca Normal (latidos por minuto)
Neonato	200 - 260
0 - 5 meses	90 - 190
6 - 12 meses	80 - 140
1 - 3 años	80 - 130
3 - 5 años	80 - 120
6 - 10 años	70 - 110
11 - 14 años	60 - 105
14 + años	60 - 100

Tomado de (Vallejo, 2011).

Tabla 5.

Pulso cardíaco en adultos

Pulso cardíaco en adultos				
Hombre	Mala	Normal	Buena	Muy buena
Edad/Pulsaciones				
20 - 29	86 o más	70 - 84	62 - 68	60 o menos
30 - 39	86 o más	72 - 84	64 - 70	62 o menos
40 - 49	90 o más	74 - 88	66 - 72	64 o menos
50 - 59	90 o más	74 - 88	68 - 74	66 o menos
60 + años	94 o más	76 - 90	70 - 76	68 o menos
Mujer	Mala	Normal	Buena	Muy buena
Edad/Pulsaciones				
20 - 29	96 o más	78 - 94	72 - 76	70 o menos
30 - 39	98 o más	80 - 96	72 - 78	70 o menos
40 - 49	100 o más	80 - 98	74 - 78	72 o menos
50 - 59	104 o más	84 - 102	76 - 82	74 o menos
60 + años	108 o más	88 - 106	78 - 88	78 o menos

Tomado de (Vallejo, 2011).

Existen diversos factores que producen un alto riesgo de sufrir enfermedades cardiovasculares y varios de estos factores no presentan algún tipo de síntoma.

- **Historial familiar:** Cierta tipo de enfermedades del corazón pueden ser hereditarias.
- **Presión alta:** El corazón realiza un mayor esfuerzo bloqueando las arterias, lo cual puede producir una angina de pecho y ataques cardíacos.
- **Colesterol elevado:** Un colesterol elevado puede producir arterias coronarias, enfermedad que puede producir un ataque cardíaco o infarto de miocardio.

Métodos y dispositivos actuales de medición.

Llevar un control de los latidos del corazón es de suma importancia en la vida de todo ser humano ya que de esta manera se evita poner en riesgo la salud humana. Existen varias formas de poder tomar el pulso y por lo tanto calcular la frecuencia cardíaca, pero por lo general se lo puede hacer por dos métodos; que son los siguientes:

- **Manualmente:** Para medir el pulso cardíaco sin la necesidad de un dispositivo electrónico se realiza el siguiente procedimiento:
 1. Colocar dedo índice y medio en una parte del cuerpo donde una arteria pase muy cerca de la piel (cuello, muñeca, ingle, sien, parte interna del codo, parte posterior de la rodilla, cara interna del pie).
 2. Una vez localizado la parte en el punto 1, presionar suavemente con los dedos y contar cuantas pulsaciones se tiene por minuto.
- **Dispositivo electrónico:** En la actualidad existen aparatos electrónicos que automáticamente realizan el proceso de medición de pulso cardíaco, el más común y utilizado es el frecuencímetro cardíaco, que se lo coloca con una cinta a la altura del pectoral. En la Figura 9 se representa la forma física del frecuencímetro cardíaco.



Figura 9. Frecuencímetro cardíaco.

Tomado de (Pecheur, s.f.)

2. CAPÍTULO II: DIMENSIONAMIENTO DEL SISTEMA DE MONITOREO

El dimensionamiento del sistema de monitoreo es uno de los aspectos principales que se debe tener en cuenta para la elaboración del proyecto, ya que tanto en el desarrollo del Software y del Hardware se deben considerar, varios parámetros como son los siguientes: sensor a utilizarse, adquisición y amplificación de la señal proveniente del sensor, transferencia de datos al dispositivo móvil y almacenamiento de la información en una base de datos.

En el presente capítulo se detallará de manera clara el estudio y análisis de los requerimientos del sistema de monitoreo; se mencionarán los componentes y elementos electrónicos que se emplearán para el diseño del prototipo, se mencionará de igual manera el aplicativo Android y el software de comparación y gestión de datos. Culminado el análisis y teniendo en claro cómo se va a realizar la parte de Software y Hardware del sistema de monitoreo, se dará paso al diseño y construcción del sistema de monitoreo, dicho proceso está detallado en el capítulo 3.

2.1. Hardware

En la parte de Hardware del sistema de monitoreo se deben tomar en cuenta las siguientes etapas:

- **Selección del sensor**
- **Etapas de amplificación y digitalización.**
- **Etapas de control:** Encendido alterno del led rojo e infrarrojo del sensor.
- **Etapas de comunicación:** Transferencia de datos a un dispositivo móvil.

En la Figura 10 se representan las etapas que serán implementadas en el diseño del prototipo.

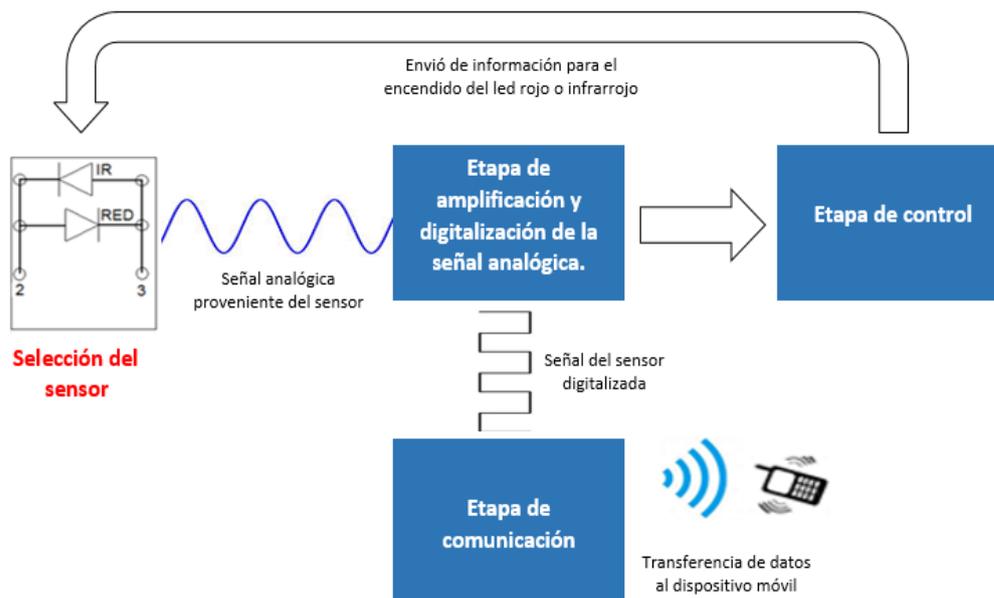


Figura 10. Diagrama de bloques del diseño del prototipo.

2.1.1. Sensor

Para seleccionar el sensor que se implementara en el prototipo se debe tomar en cuenta ciertos parámetros como son: rango de medición en la saturación de oxígeno en la sangre y pulso cardíaco, el tiempo de respuesta, tipo de respuesta y el tipo de señal que se obtiene a la salida, es tos parámetros se detallan en la tabla 6.

El sensor que se implementará en el sistema de monitoreo deberá ser capaz de medir la saturación de oxígeno en la sangre y el pulso cardíaco, de manera precisa y correcta, adicional se necesita que sea de fácil conexión con el prototipo a diseñarse, por último, se requiere que el sensor sea compatible con las etapas de amplificación – digitalización, control y comunicación; con el fin de que la señal detectada por el sensor puede ser procesada y enviada al dispositivo móvil.

Existen varias marcas y modelos ofertados por los fabricantes para la fabricación de sensores de Spo2 (saturación de oxígeno presente en la sangre); para un análisis comparativo del sensor más adecuado se ha escogido 3 de las marcas más comercializadas las cuales son: Nellcor, Datascope y Philips.

Tabla 6.

Característica de los sensores.

	Nellcor	Datascope	Philips
Rango de medición de spo2	(1 – 100)% Resolución: ± 2 %	(0 – 98)% Resolución: ± 2 %	(0 - 100)% Resolución: ± 2 %
Rango de medición de bpm(pulsos por minuto)	(20 - 250) bpm Resolución: ± 1 bpm	(25 - 240) bpm Resolución: ± 1 bpm	(30 - 300) bpm Resolución: ± 1 bpm
Tiempo de respuesta	400 ms.	300 ms.	350 ms.
Tipo de conector	DB9	N Datex conector	Pin macho Merlin
Señal de salida	Analógica	Analógica	Analógica

Tomado de (Covidien, Cables y sensores. s.f.)

Comparando los datos de la tabla 6 se ha optado por el sensor del fabricante Nellcor modelo DS-100A, la gran ventaja del sensor Nellcor DS-100A frente a otros sensores es que este sensor usa la tecnología OXIMAX para proporcionar funciones lecturas de gran exactitud y fiabilidad en condiciones adversas como por ejemplo en casos de baja perfusión o en situaciones de interferencias de la señal, donde se pueden presentar problemas en la adquisición de los datos.



Figura 11. Sensor Nellcor.

Tomado de (Cables y sensores, s.f.)

El sensor Nellcor modelo DS-100A está indicado para la vigilancia continua no invasiva de la saturación de oxígeno arterial y la frecuencia del pulso en pacientes de más de 40 kg de peso. (Nellcor, s.f.).

En la tabla 7 se presentan las principales características del sensor Nellcor DS – 100A.

Tabla 7.

Características del sensor Nellcor DS – 100A

Intervalos de medición	
Intervalo de saturación de SPO2	(1 - 100) %
Intervalo de frecuencia de pulso	(20 - 250) latidos por minuto (lpm)
Intervalo de perfusión	(0.03 - 20)%
Exactitud de las medidas	
Exactitud de la frecuencia de pulso.	(20 - 250) latidos por minuto (bpm).
SpO2 exactitud en la saturación	(70 - 100) % de (+/- 2 - +/- 3) dígitos.
Intervalo de funcionamiento y disipación	
Longitud de onda de luz roja	Aproximadamente 660 nm
Longitud de onda de luz infrarroja	Aproximadamente 990 nm
Potencia de salida óptica	< 15 Mw (megawatt)
Disipación de alimentación	52.5 Mw (megawatt)
Características eléctricas	
Consumo de voltaje	5 V
Consumo de corriente	< 40 Ma (Mega-amperio)

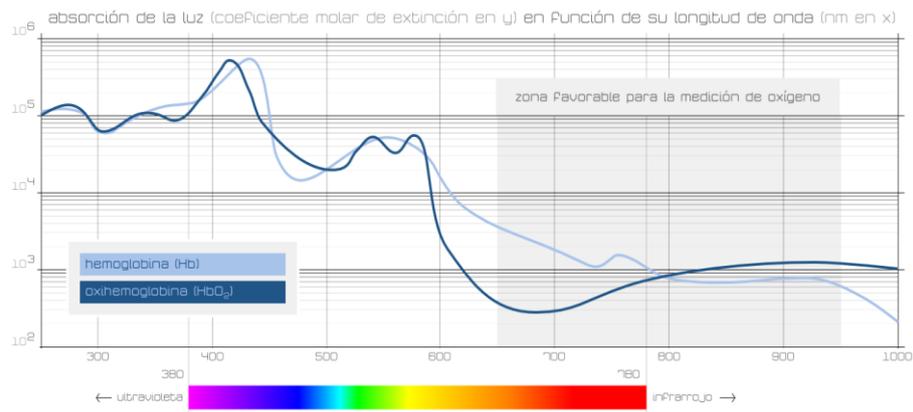
Tomado de. (Covidien. s.f.)

Funcionamiento del sensor.

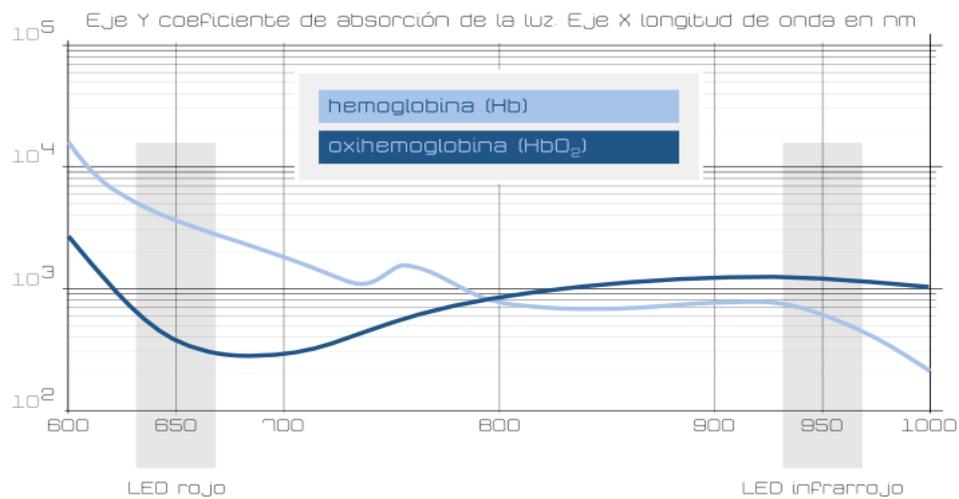
Todos los sensores destinados para la medición de la saturación de oxígeno en la sangre proporcionan una medición no invasiva y continúa del porcentaje de hemoglobina oxigenada, y a su vez estos sensores también proporcionan datos acerca de la frecuencia o pulso cardíaco. Cuando la sangre pasa por los pulmones se oxigena, y la hemoglobina puede llevar oxígeno convirtiéndose de esta manera en oxihemoglobina.

Para la medición no invasiva del oxígeno transportado por la hemoglobina, los sensores se basan en la ley de absorción óptica de Lambert-Beer, la cual establece que la concentración de una sustancia puede determinarse por la absorción de luz. (Cenetec. S.f.).

Tanto la hemoglobina como la oxihemoglobina tienen diferentes niveles de absorción para los diferentes niveles de onda, en la Figura 12 se representa lo anteriormente escrito.



a) Absorción de la luz en función de su longitud de onda.



b) Relación de la hemoglobina y oxihemoglobina.

Figura 12. Funcionamiento de un sensor de oximetría. a) Absorción de la luz en función de su longitud de onda. b) Relación de la hemoglobina y oxihemoglobina. Tomado de (Ventura, s.f.)

Como se puede observar en la Figura 12 a) y 12 b) hasta alrededor de los 800 nm la hemoglobina absorbe mayor cantidad de luz roja especialmente en los 660 nm y a partir de ese valor en adelante la oxihemoglobina absorbe más luz infrarroja especialmente en los 905 nm. Basándose en ese comportamiento, el sensor Nellcor modelo DS-100A está compuesto por 2 leds: un rojo con una longitud de onda de 660 nm. y un led infrarrojo con una longitud de onda de 905 nm, adicional el sensor posee un fotorreceptor para poder hacer la lectura simultánea de la transmisión de las longitudes de onda.

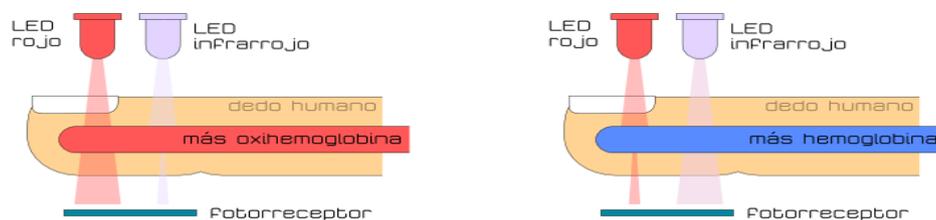


Figura 13. Esquema del funcionamiento del sensor Nellcor DS-100A.

Tomado de (Ventura, s.f.)

En la Figura 13, se observa que la longitud de onda producida por el led rojo es absorbida en su mayoría por la hemoglobina, y la longitud de onda producida por el led infrarrojo es absorbida en su mayoría por la oxihemoglobina. El funcionamiento del sensor se asemeja a un transductor, es decir entra una señal fisiológica y a la salida se obtiene una señal eléctrica que representa la variación de cambio entre la sangre con oxígeno (oxihemoglobina) y sin oxígeno (hemoglobina) y está en el orden de los 0 – 350 mv., luego esa señal es amplificada y digitalizada en la etapa de amplificación y digitalización.



Figura 14. Conector DB9 hembra del sensor Nellcor.

Tomado de (Suatunce, 2014)

La descripción de cada uno de los pines del sensor se muestra en la tabla 8.

Tabla 8.

Descripción de los pines del sensor

Pin	Descripción
1	Resistencia del sensor
2	Polarización de los leds
3	Polarización de los leds
4	No tiene conexión
5	Ánodo del fotodiodo (Salida +)
6	Resistencia del sensor
7	Tierra del cable
8	No tiene conexión
9	Cátodo del fotodiodo (Salida -)

Tomado de (Suatunce, 2014)

2.1.2. Etapa de amplificación y digitalización.

La señal que se obtiene a la salida de sensor Nellcor DS-100A, está en el orden de 0 – 350 mv., por lo que se requiere que esta señal sea acondicionada y amplificada. El Atmega164P es un micro controlador CMOS de 8 bits de bajo consumo basado en la arquitectura RISC (Reduced Instruction Set Computer) mejorada, cuyas instrucciones se ejecutan en un ciclo de máquina, el

ATmega164P consigue transferencia de información alrededor de 1 MIPS (Atmel, s.f.).

Gracias a las ventajas y características que posee el micro controlador Atmega164P se lo ha escogido para que sea implementado en el prototipo; especialmente por el tamaño en SMD (tecnología de montaje superficial) que ahorra espacio y permite que el circuito sea lo más pequeño posible, otra de las razones por lo que se optó por el micro controlador Atmega 164P es por las características de amplificación diferencial, dicha característica permite amplificar los voltajes en el orden de los mV, sin necesidad de amplificadores adicionales, ya que internamente se puede configurar esa señal para que sea multiplicada por una ganancia (1x – 0dB, 10x – 20 dB, 200x – 46 dB).

Como se mencionó anteriormente la salida de voltaje del sensor Nellcor DS – 100 A en los pines 5 y 9 (Figura 14), está en el rango de 0 - 350 mv., esa señal obtenida por el sensor ingresa al microcontrolador Atmega164 p, el cual cuenta internamente con todos los componentes de amplificación y luego realiza la conversión analógica digital, los valores obtenidos de esta conversión varían en el rango de 0 – 402, y representan la señal digitalizada de la variación de cambio entre la sangre con oxígeno (oxihemoglobina) y sin oxígeno (hemoglobina).

Características.

Entre las principales características del micro controlador Atmega 164 P se puede destacar las siguientes las cuales detalladas en la tabla 9:

Tabla 9.

Características del micro controlador Atmega 164P.

Arquitectura Avanzada RISC
131 instrucciones que se ejecutan en un ciclo de reloj.
32 registros de trabajo de 8 bits para propósito general.
Capacidad de procesamiento de unos 20 MIPS a 20 MHz.
Memorias de programa y de datos no volátiles de alta duración
16 Kbytes de memoria FLASH.
512 bytes de memoria EEPROM.
1 Kbyte de memoria SRAM.
Características de los periféricos
Convertidor analógico digital con una resolución de 10 bits.
Ganancia diferencial seleccionable entre (x1, x10 y x200)
Comparador Analógico dentro del mismo Chip
Líneas de entradas y salidas
32 líneas programables de entradas y salidas
Voltajes de Operación
(2,7 – 5,5) V.
Velocidad de funcionamiento
(0 – 20) Mhz.

Tomado de (Atmel, 2014)

La distribución de los pines del micro controlador Atmega 164P, se muestra en la Figura 15.

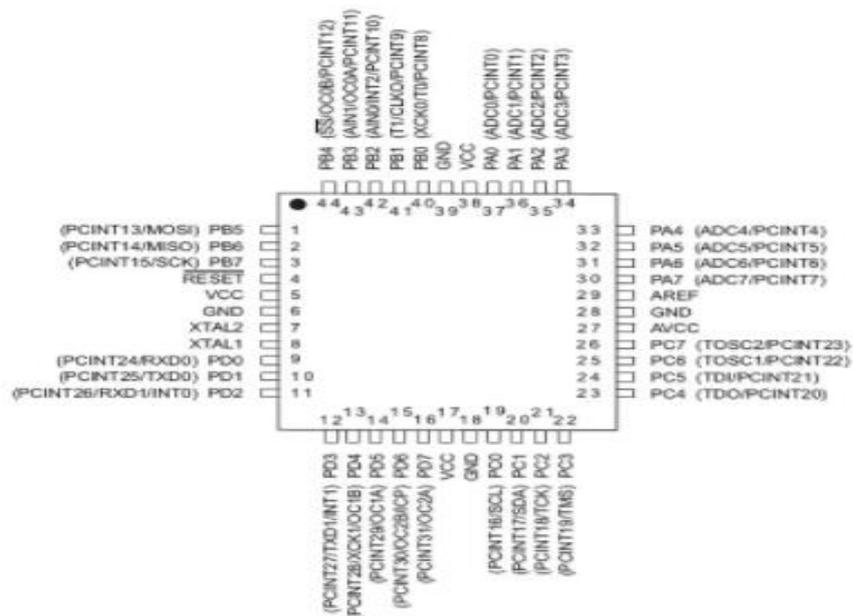


Figura 15. Distribución de pines del micro controlador Atmega 164 P.
Tomado de (Atmel, s.f.)

Tabla 10.

Descripción de los pines del micro controlador Atmega 164 P.

Pin	Descripción
GND	Conexión a tierra.
Puerto A	Entradas analógicas para el convertor análogo/digital.
(A0 y A1)	A0 = Entrada diferencial negativa ; A1 = Entrada diferencial positiva
Puerto B	Programación del micro controlador.
(B5, B6 y B7)	B5 = MOSI; B6 = MISO y B7 = SCK
Puerto D	Comunicación serial con el Bluetooth HC - 06
(D0, D1, D3 y D5)	D0 = RX y D1 = Tx; D3 y D5 = Tiempo para el encendido leds
AREF	Referencia de voltaje para el convertor análogo/digital.
XTAL1	Pines para conexión del oscilador.
XTAL2	Pines para conexión del oscilador.

Tomado de (Atmel, s.f.).

2.1.3. Etapa de control.

Debido a la conexión interna en el sensor de los leds rojo e infrarrojo, la cual está representada en la Figura 16, se requiere de alguna manera elaborar una fase de polarización para que el fotodetector pueda captar las señales emitidas.

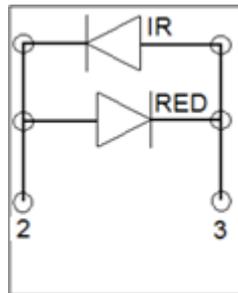


Figura 16. Conexión interna de los leds del microcontrolador Atmega 164 P. Tomado de (Ronquillo, 2013)

Como se puede observar en la Figura 16 los leds rojo e infrarrojo están colocados inversamente, con el fin de que al conmutar la polarización se encienda uno u otro, es por este motivo que se debe realizar una etapa de control que realice las siguientes funciones:

- Establecer el tiempo de encendido de cada led.
- Determinar e indicar que led se está encendiendo.

Para el diseño de la etapa de control, se ha optado por implementar el integrado L293D, el cual internamente posee 4 circuitos con 4 salidas push-pull (circuito electrónico que puede impulsar una corriente eléctrica positiva o negativa en una carga), este integrado tiene la capacidad de controlar corrientes de hasta 600 mA y con tensiones que van desde los 4.5V hasta los 36V, en la Figura 17, se representa la forma física del integrado L293D.

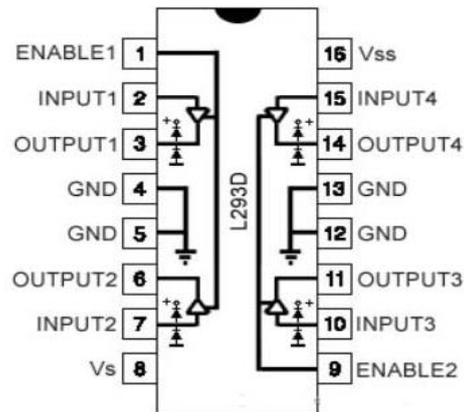


Figura 17. Integrado L293D y su configuración de pines.

Tomado de (Prometec, s.f.)

La descripción de los pines del integrado L293D se muestra en la tabla 11.

Tabla 11.

Descripción de los pines del integrado L293D

Pin	Descripción
1	Habilitación del puente 1.
2	Entrada del amplificador 1.
3	Salida del amplificador 1.
6	Salida del amplificador 2.
7	Entrada del amplificador 2.
8	Fuente de alimentación (motores).
9	Habilitación del puente 2.
10	Entrada del amplificador 3.
11	Salida del amplificador 3.
14	Salida del amplificador 4.
15	Entrada del amplificador 4.
16	Fuente de alimentación (lógica).

Tomado de (Tecnikro, s.f.).

Los leds del sensor tanto rojo como infrarrojo se encenderán por un tiempo de 4 segundos, dicho tiempo será establecido en el micro controlador Atmega164P, para que comunique los pulsos altos y bajos, al Integrado L293D que es el encargado de prender y apagar alternadamente los leds, el proceso anteriormente descrito será explicado en el capítulo 3 sección 3.3.

2.1.4. Etapa de comunicación.

Para el diseño de la etapa de comunicación se ha optado por utilizar un módulo Bluetooth Slave (HC-06). El módulo Bluetooth HC-06 se trata de un dispositivo económico y sencillo de utilizar, por lo general está configurado de fábrica para que opere como esclavo, es decir, preparado para escuchar peticiones de conexión, la ventaja de este módulo es que puede ser controlado a una distancia de 10 m con un celular, una tablet o una laptop. En la Figura 18 se representa la forma física del módulo Bluetooth HC – 06.

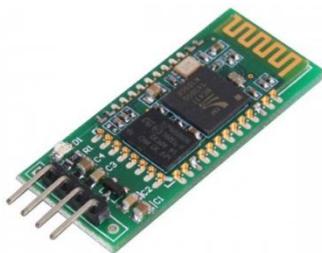


Figura 18. Módulo Bluetooth HC – 06.

Tomado de (BotScience, s.f.)

Características.

Entre las principales características del módulo Bluetooth HC – 06 se pueden destacar las siguientes:

- Compatible con el protocolo Bluetooth V2.0.
- Voltaje de alimentación: 3.3VDC – 6VDC.
- Voltaje de operación: 3.3VDC.
- Corriente de operación: < 40 mA

2.1.5. Fuente de alimentación.

La mayoría de dispositivos electrónicos necesitan para su funcionamiento proveerse de tensión continua, para el presente proyecto se implementan 2 baterías Samsung B500AE. Las principales características de la batería Samsung están representadas en la tabla 12.

Tabla 12.

Características batería Samsung B500AE.

Detalle	Descripción
Tecnología	Li-ion
Capacidad	1900 mAh, 3.8 voltios

Tomado de (Samsung, s.f.)

Debido a que el módulo Bluetooth necesita un voltaje de 5 V para su operación, se implementarán 2 baterías conectadas en serie teniendo como resultado una tensión 7.6 V, pero como se mencionó anteriormente se necesita una tensión de 5 V, por tal motivo es necesaria una etapa de regulación de voltaje para lo cual se implementa el circuito integrado regulador L7805CV que regula el voltaje a 5 V. Las características del circuito integrado regulador L7805CV se presentan en la tabla 13.

Tabla 13.

Características del circuito integrado regulador L7805CV

Detalle	Descripción
Voltaje de salida	5 V.
Corriente de salida	1.5 A.
Voltaje de entrada máximo	35 V.
Regulación de línea	100 mV.
Voltaje de entrada mínimo	7 V.

Tomado de (MouserElectronics, s.f.)

2.2. Software

Las aplicaciones a diseñarse tanto para el dispositivo móvil como para el diseño del software de comparación y gestión de datos, deben procesar la información obtenida por la etapa de acondicionamiento, digitalización y comunicación a una interfaz gráfica que sea adaptable y de fácil uso tanto para el paciente como para el médico tratante.

La versión del aplicativo móvil será desarrollada para el sistema operativo Android de tal manera que el paciente pueda obtener notificaciones de alerta en su dispositivo móvil cuando existe alguna variación en la saturación de oxígeno en la sangre, mientras que el software para la comparación y gestión de datos, será desarrollado en el lenguaje de programación C#.

Tanto el aplicativo Android como el software para la comparación y gestión de datos podrán interactuar y almacenar los datos que se consideren necesarios en una base de datos diseñada en SQL Server. Para que la base de datos, el aplicativo Android y el software de comparación y gestión de datos, intercambien información a través del internet se necesita de un alojamiento web el cual será Somee.com.



Figura 19. Esquema general del sistema

2.2.1. Base de datos.

El sistema de monitoreo y control requiere de alguna manera almacenar los datos del Médico con el fin de que pueda ingresar al software de comparación y gestión de datos y de igual manera el sistema de monitoreo necesita registrar los datos provenientes de la etapa de comunicación que provoquen la activación del mensaje de alerta en el dispositivo móvil, para que de esta manera el Médico de cabecera puede tener acceso a ellos, es por eso que es necesario contar con un sistema de gestión de base de datos para poder cumplir este requerimiento.

La base de datos será diseñada en SQL server que es un sistema de administración y análisis de base de datos relacional el cual permite el almacenamiento de datos e información. (Mundo informático, s.f.).

Entre las principales características de SQL server se pueden mencionar las siguientes:

- Integración con internet.
- Escalabilidad y disponibilidad.
- Permite trabajar en modo cliente-servidor.
- Permite administrar información de otros servidores.
- Incluye un entorno gráfico de administración.

2.2.2. Aplicativo Android.

Android es un sistema operativo pensado para teléfonos inteligentes, que permite desarrollar y programar aplicaciones, además Android proporciona todas las interfaces para poder desarrollar aplicaciones que accedan a las funciones de un teléfono móvil como, por ejemplo: GPS, llamadas, agenda, entre otras. (Nieto, 2011).

El aplicativo para el dispositivo móvil se lo realizará con Eclipse Kepler este software contiene un conjunto de herramientas para poder realizar o desarrollar software, de igual manera permite a los programadores poder realizar aplicaciones con una interfaz gráfica amigable para el usuario.

Características y ventajas de Android.

Como se mencionó anteriormente, Android es un sistema operativo abierto y de multitarea, que permite a los programadores y usuarios realizar y acceder a diversas aplicaciones y funcionalidades de un dispositivo móvil. (Android OS. s.f).

Entre sus características y ventajas se puede destacar las siguientes.

- Código abierto.
- Núcleo basado en el Kernel de Linux.
- Adaptable a muchas pantallas y resoluciones.
- Utiliza SQLite para el almacenamiento de datos.
- Ofrece diferentes formas de mensajería.
- Navegador web basado en WebKit incluido.
- Soporte de Java y muchos formatos multimedia.
- Soporte de HTML, HTML5, Adobe Flash Player, etc.

2.2.3. Software de comparación y gestión de datos.

Para la comparación y gestión de los datos se empleará un aplicativo diseñado en el lenguaje de programación C#, que conjuntamente con una base de datos creada SQL server, facilitará al médico tratante tener un acceso a todos los eventos y registros del paciente para de esta manera poder realizar los estudios y controles respectivos.

- **Ventajas y desventajas.**

C# es un lenguaje de programación orientado a objetos, programar en C# tiene varias e importantes ventajas sobre otros lenguajes de programación.

Para el desarrollo del aplicativo de comparación y gestión de datos del sistema de monitoreo, se tomó como referencia 3 lenguajes de programación que son C++, java y C#.

Tabla 14.

Tabla comparativa de los lenguajes de programación.

	Ventajas	Desventajas
C#	<ul style="list-style-type: none"> - Editor gráfico integrado. - Facilidad de uso y considerable capacidad para acelerar el tiempo de desarrollo. - Permite realizar aplicaciones web. 	<ul style="list-style-type: none"> - Se debe obtener una versión reciente de Visual Studio. - Se ve obligado a usar Framework.net.
C++	<ul style="list-style-type: none"> - Se puede rehusar el código. - Genera programas compactos. 	<ul style="list-style-type: none"> - No es atractivo visualmente. - No soporta la creación de páginas web.
Java	<ul style="list-style-type: none"> - Fácil aprendizaje. - Funciona con las principales plataformas de hardware y sistema operativos. 	<ul style="list-style-type: none"> - Lentitud en la ejecución de los programas - Algunas herramientas tienen un costo adicional.

Tomado de: (Manosalvas, s.f.)

Comparando la información de la tabla 14 se optó por implementar C# como el lenguaje de programación, para el diseño del software de comparación y gestión de los datos, ya que las desventajas que posee no impiden de ninguna manera el funcionamiento del sistema de monitoreo y además C# permite y soporta el diseño de páginas web.

Entre las principales características de C# se pueden destacar las siguientes:

- Facilidad de uso.
- Programación orientada a objetos.
- Administración de memoria.
- Seguridad en el manejo de los datos.
- Uso de operadores.
- Compatibilidad.

2.2.4. Alojamiento web.

Para el funcionamiento correcto del sistema de monitoreo es necesario un alojamiento web donde se pueda almacenar la base de datos diseñada en SQL Server, para que, de esta manera, tanto médico como paciente interactúen con dicha base a través de internet, por tal motivo se optó por usar un alojamiento web en Somee.com.

- **Somee.com:** Es un alojamiento web que ofrece el poder y la flexibilidad de la última tecnología de Windows, IIS (Internet Information Services), Microsoft SQL Server y una amplia variedad de otras características de alojamiento como ASP y ASP.Net.

3. CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE MONITOREO

El diseño es un paso primordial en el proceso de desarrollo de cualquier dispositivo o sistema electrónico; cuyo objetivo es obtener un modelo representativo de lo que se va a construir. En el presente capítulo se describe el diseño e implementación del prototipo del sistema de monitoreo, se realizará un mayor énfasis en las etapas de Hardware anteriormente mencionadas en el capítulo II, de igual manera serán analizados más a fondo el aplicativo móvil, el diseño del software para el control y el gestionamiento de datos y la base de datos diseñada para el proyecto.

3.1. Sensor Nellcor DS – 100A.

El sensor Nellcor DS – 100 A transforma la señal fisiológica a una señal eléctrica, luego esta señal eléctrica ingresa al micro controlador Atmega para ser procesada. La señal obtenida en los pines 5 y 9 a la salida del sensor tanto del led rojo como del led infrarrojo está representada en la Figura 20.

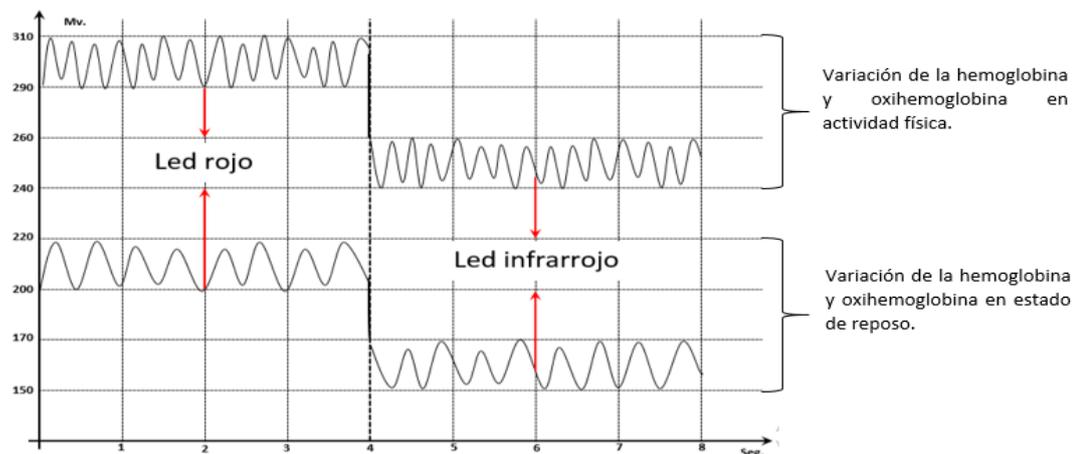


Figura 20. Señal obtenida a la salida del sensor Nellcor DS – 100 A.

En la Figura 20 se puede divisar que existen unos picos de voltaje los cuales representan la variación de cambio entre la sangre con oxígeno (oxihemoglobina) y sin oxígeno (hemoglobina), esa señal ingresará al micro controlador Atmega164 p, para la etapa de amplificación y digitalización.

La cantidad de pulsos que el corazón puede dar por minuto cuando está en estado de reposo, se encuentran detalladas en la tabla N.- 15.

Tabla 15.

Cantidad de pulsos que el corazón puede dar por minuto

Detalle	Latidos por minuto
Recién nacido	80 – 180
1 año	80 – 140
10 años	50 – 90
Adolescentes	50 – 90
Adultos y ancianos	60 – 100

Tomado de: (Urespuestas, 2015)

El sistema de monitoreo y control está destinado para personas adultas y de tercera edad por tal motivo se tomará como referencia los datos correspondientes a esas personas, adicional cabe recalcar que el rango de la frecuencia cardíaca en adultos puede aumentar de 150 a 200 latidos por minuto durante la actividad física. Los leds del sensor se encenderán alternadamente cada 4 segundos obteniendo cierta cantidad de picos de voltaje, el número de pico de voltajes que se obtiene en 4 segundos viene relacionada de la siguiente manera: La cantidad de pulsos que se obtiene por minuto de una persona adulta está en el rango de 60 – 200 latidos, los picos de voltaje representan la variación de la hemoglobina y oxihemoglobina; las cuales dependen directamente de los latidos del corazón, por tal razón los picos de voltaje a la salida del sensor están relacionados con los latidos del corazón.

Para determinar el número de picos que se obtiene a la salida del sensor cada 4 segundos, se toma la cantidad mínima y máxima de latidos que una persona puede llegar a tener en un minuto y se implementa una regla de tres simple.

- Valor mínimo.

60 segundos \longrightarrow 60 latidos

4 segundos \longrightarrow X

X = 4 picos de voltaje en 4 segundos

- Valor máximo.

60 segundos \longrightarrow 200 latidos

4 segundos \longrightarrow X

X = 13,33 \approx 13 picos de voltaje en 4 segundos

La cantidad de picos de voltaje que se puede obtener esta en el rango de 4 – 13 picos, los cuales pasarán a la etapa de amplificación y digitalización.

3.2. Etapa de amplificación y digitalización.

La etapa de amplificación y digitalización está dada por el micro controlador Atmega164P, el cual tiene una etapa de amplificación diferencial y un convertidor analógico – digital (ADC). El (ADC) del Atmega 164P posee las siguientes características:

- Convertidor analógico - digital con una resolución de 10 bits.
- Un tiempo de conversión aproximado de (13 a 260) μ s
- 15.000 muestras/segundo en su resolución máxima.
- Un total de 7 canales para las entradas diferenciales.
- 0 V - Vcc para el voltaje de entrada

- La conversión puede ser única o libre.
- Inicialización del convertidor analógico - digital mediante un auto disparo con fuente de interrupción.

3.2.1. Amplificación diferencial.

En el Atmega164P las entradas diferenciales (ADC1, ADC0 y ADC3, ADC2), poseen una ganancia que puede ser programada, otorgando amplificaciones de 0db, 20db y 46db, esta ganancia amplifica el voltaje de entrada diferencial, para la realización del prototipo se usa las entradas diferenciales (ADC1, ADC0) con una ganancia programable de 20 db (10x).

El registro ADMUX, es un registro destinado para el control del convertidor analógico – digital, el cual está conformado por 8 bits, en la implementación del prototipo el registro ADMUX define el valor de la ganancia y el voltaje de referencia, en la Figura 21 se representa la estructura del registro ADMUX.

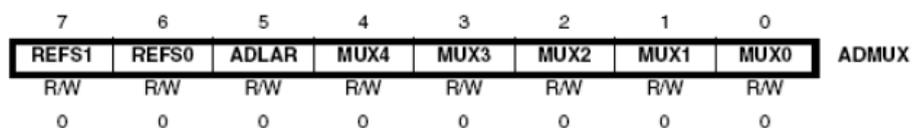


Figura 21. Estructura del registro ADMUX.

Tomado de (Micro controladores, s.f.)

La descripción de los bits se detalla en la tabla 16.

Tabla 16.

Descripción de los Bits del registro ADMUX.

Bit	Función
Bit 7:6 – REFS1:0	Seleccionar el voltaje de referencia.
BIT 5 – ADLAR	Alinear los bits a la izquierda
BIT 4:0 – MUX4:0	Seleccionar la ganancia y la entrada analógica.

Tomado de (Micro controladores, s.f.)

La combinación de entrada analógica y la ganancia diferencial (Gain) son seleccionadas con la combinación de los BIT MUX 4:0 del registro ADMUX representado en la Figura 21, las opciones de entradas analógicas y la ganancia diferencial que se pueden seleccionar se detallan en la tabla 17.

Tabla 17.

Combinación de los Bits MUX4:0 del registro ADMUX.

MUX4..0	Una sola entrada	Entrada diferencial positiva	Entrada diferencial negativa	Ganancia
00000	ADC0			
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
01000	NA	ADC0	ADC0	10x
01001		ADC1	ADC0	10x

NA

Tomado de: (Micro controladores, s.f.)

La entrada analógica y la ganancia diferencial, implementadas para el diseño e implementación del sistema de monitoreo y control está dada por la siguiente combinación de bits:

Tabla 18.

Combinación de los Bits en el registro ADMUX

MUX4	MUX3	MUX2	MUX1	MUX0
0	1	0	0	1

Tomado de: (Atmel, s.f.).

Si se busca la combinación de los bits de la tabla 18, en la tabla 17, se puede observar que el sistema de monitoreo para la amplificación diferencial trabaja de la siguiente manera:

- **Entrada diferencial positiva:** Pin ADC1 del Atmega 164P
- **Entrada diferencial negativa:** Pin ADC0 del Atmega 164P
- **Ganancia (Gain):** Se tiene una ganancia de 10 para la amplificación.

3.2.2. Convertidor analógico – digital (ADC).

Una vez que las entradas análogas son seleccionadas, la ganancia diferencial amplifica la diferencia de voltaje de las entradas análogas, este valor amplificado es la entrada análoga del ADC del Atmega164P. “El ADC del Atmega164P, posee un circuito de muestreo y retención asegurando de esta manera que a la entrada del convertidor analógico-digital exista un voltaje constante” (Atmel, s.f.).

Como se puede observar en las características del ADC del Atmega164P posee un voltaje de referencia interno que comprende entre 2.5 V o 1.1 V, pero también se puede tomar un voltaje de referencia externo. Con los bits RFS1 y RFS0 del registro ADMUX, se seleccionan el voltaje de referencia para el ADC como se muestra en la tabla 19.

Tabla 19.

Combinación de los Bits REFS1 y REFS0 del registro ADMUX

REFS1	REFS0	Selección del voltaje de referencia
0	0	AREF, voltaje de referencia interno apagado
0	1	AVCC con capacitor externo en el pin AREF
1	0	Voltaje de referencia interno 1.1V con capacitor en el pin AREF
1	1	Voltaje de referencia interno 2.56V con capacitor en el pin AREF

Tomado de (Atmel, s.f.).

Para la implementación del prototipo se ha optado por escoger la combinación (0 0) en REFS1 y REFS2, es decir se trabaja con un voltaje de referencia (v_{ref}) externo de 5 V. Con un v_{ref} de 5 V. se obtienen los siguientes datos:

- **Entrada diferencial:** Para calcular el valor máximo de entrada diferencial que se puede tener en las entradas analógicas diferenciales del Atmega164 P, se debe tomar en cuenta que a la salida del ADC solo se puede tener valores en el rango de -512 a + 511. (Atmel, s.f.).

En la gráfica 20 se observa que la onda obtenida por el sensor no tiene valores negativos, por ende, si no se tiene valores negativos a la entrada no se obtendrá valores negativos a la salida, por tal motivo para calcular el máximo valor que se puede obtener en la entrada del ADC se toma como referencia el máximo valor positivo que se obtiene a la salida del ADC que es 511 y se procede a calcular con la siguiente fórmula.

$$ADC = \frac{(V_{pos} - V_{neg}) \cdot Gain \cdot 512}{V_{ref}} \quad (\text{Ecuación 1})$$

$$(V_{pos} - V_{neg}) = \frac{(511)(5V)}{(10) \cdot (512)}$$

$$(V_{pos} - V_{neg}) = 499 \text{ mV}$$

Con una ganancia de 10 en la entrada diferencial; se puede medir hasta 499 mv, y si se observa la Figura 20 el rango de la señal obtenida por el sensor Nellcor DS – 100A es de (0 – 310) mv.

- **Muestreo de la señal.**

Anteriormente se demostró que, en 4 segundos, el led rojo y el infrarrojo pueden producir entre 4 – 13 picos de voltaje, si el mismo procedimiento se aplica para 1 segundo se obtiene el siguiente resultado:

- Valor mínimo.

60 segundos \longrightarrow 60 latidos

1 segundos \longrightarrow X

X = 1 picos de voltaje en 1 segundo

- Valor máximo.

60 segundos \longrightarrow 200 latidos

1 segundos \longrightarrow X

X = 3,33 \approx 4 picos de voltaje en 1 segundo

La cantidad de picos de voltaje que se obtiene en un segundo para el led rojo e infrarrojo está entre 1 – 4, para que esta señal análoga pueda ser reconstruida, se debe hacer referencia a la teoría de muestreo de Nyquist que establece lo siguiente “Para que una señal sea reconstruida su frecuencia de muestreo debe ser 2 veces de la señal a muestrear”. (Scribd. s.f.).

Para el diseño del prototipo se ha optado por muestrear a una frecuencia de 14 muestras/segundo, lo cual es suficiente para poder procesar la señal original, si a las 14 muestras/segundo se las multiplica por los 4 segundos que es el tiempo que permanecerán encendido cada led, se obtiene 56 muestras/4 segundos tanto para el led rojo como para el led infrarrojo.

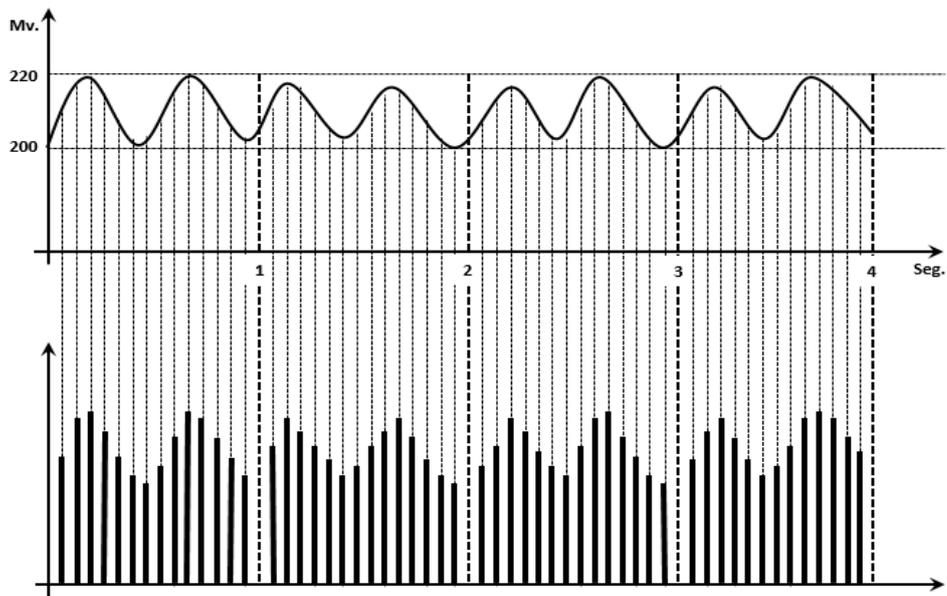


Figura 22. 56 muestras/4 segundos correspondientes al led rojo

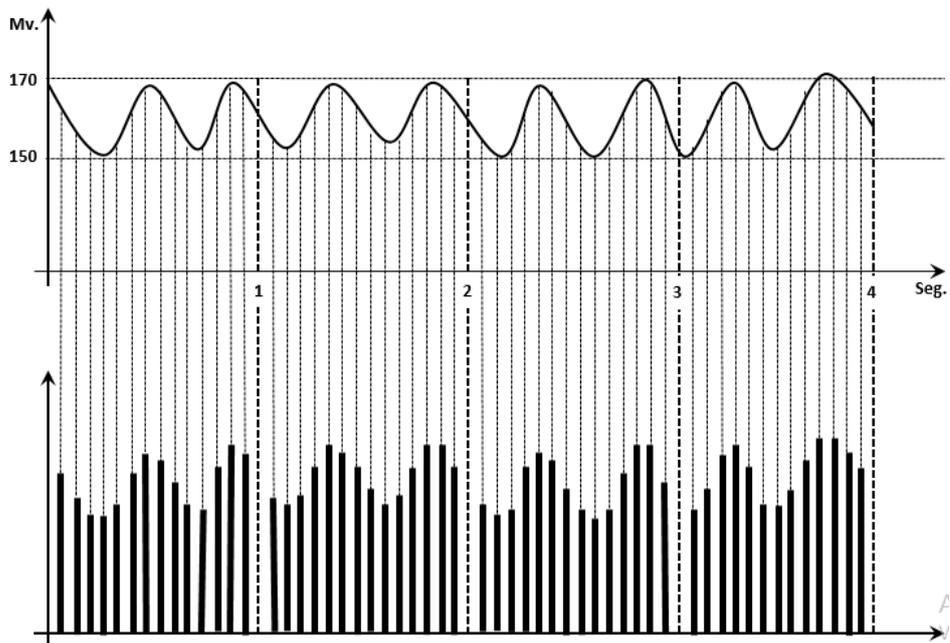


Figura 23. 56 muestras/4 segundos correspondientes al led infrarrojo

Para la obtención del pulso cardíaco y la saturación de oxígeno en la sangre se deben analizar las muestras, tanto del led rojo e infrarrojo, y seguir el proceso, que se detallará en el presente capítulo sección 3.7.

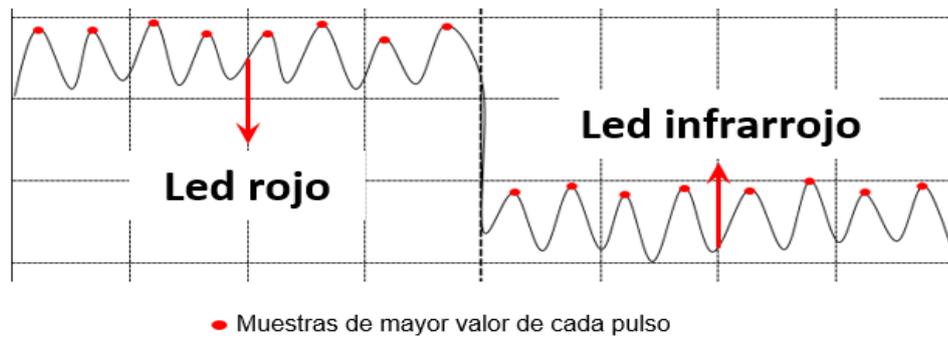


Figura 24. Pulsos correspondientes al led rojo e infrarrojo

En las Figuras 25, 26 y 27 se muestran los diagramas de flujo del proceso anteriormente descrito.

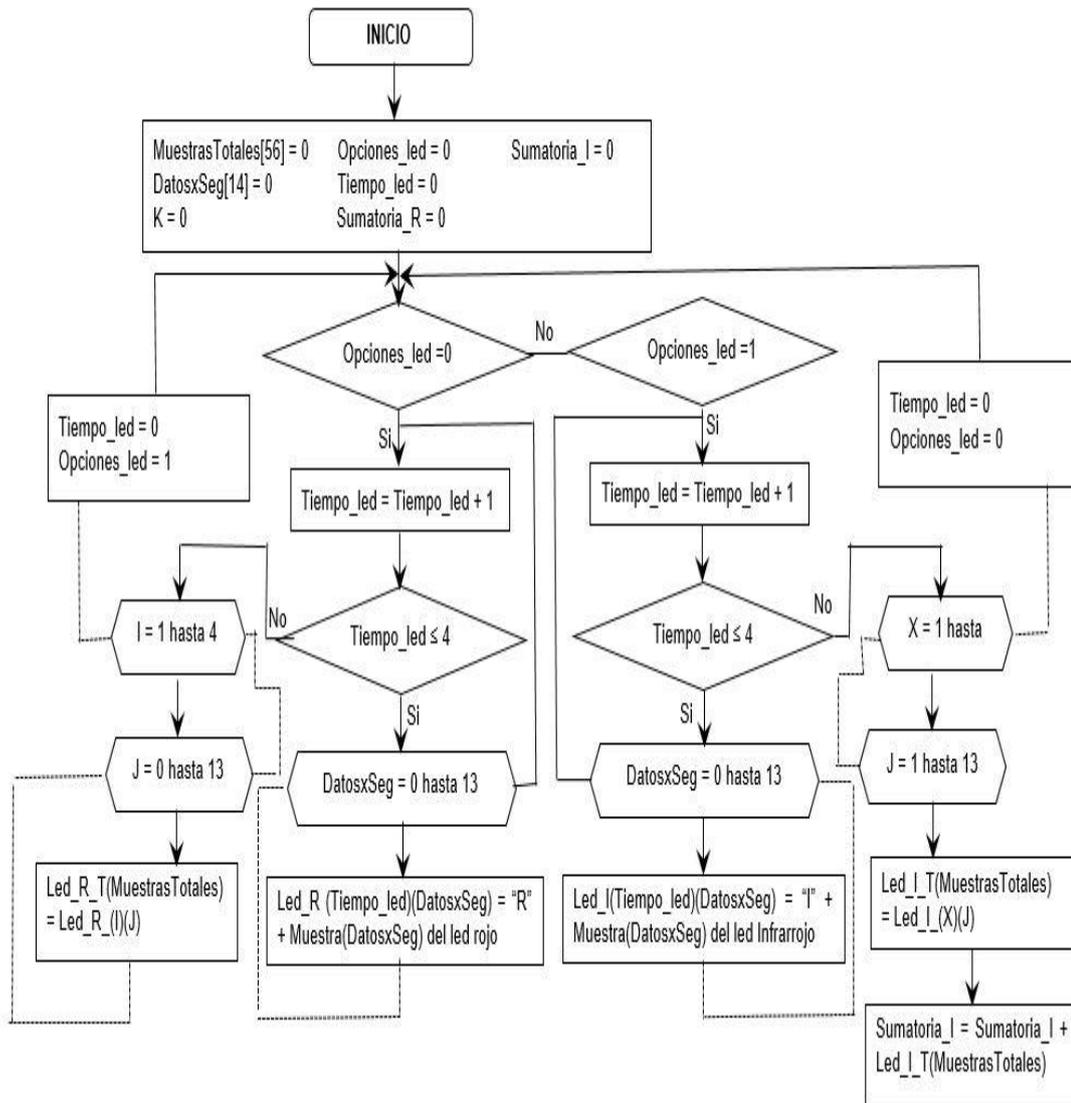


Figura 25. Encendido del led rojo e infrarrojo

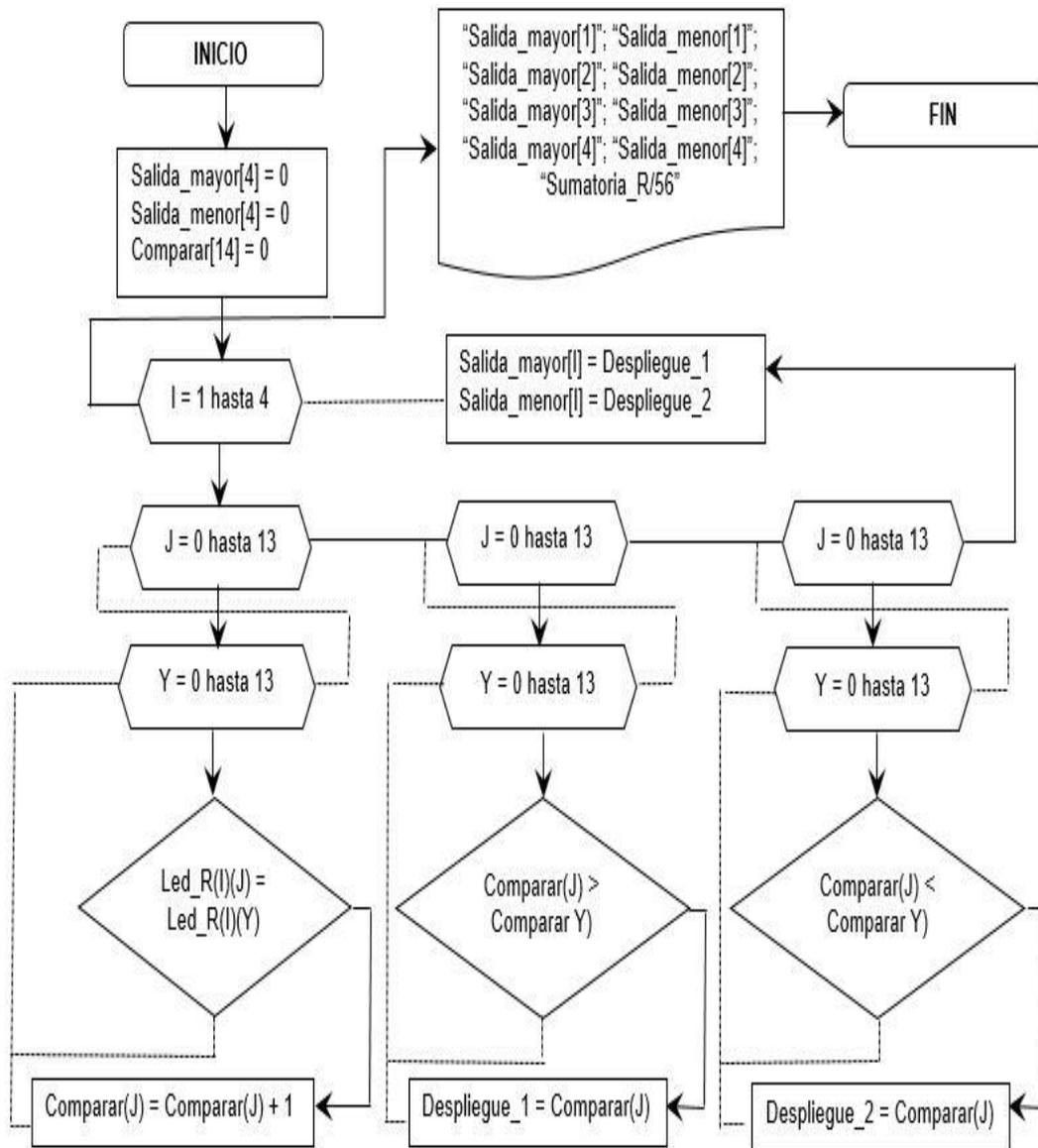


Figura 26. Procesamiento de muestras del led rojo.

3.3. Etapa de control.

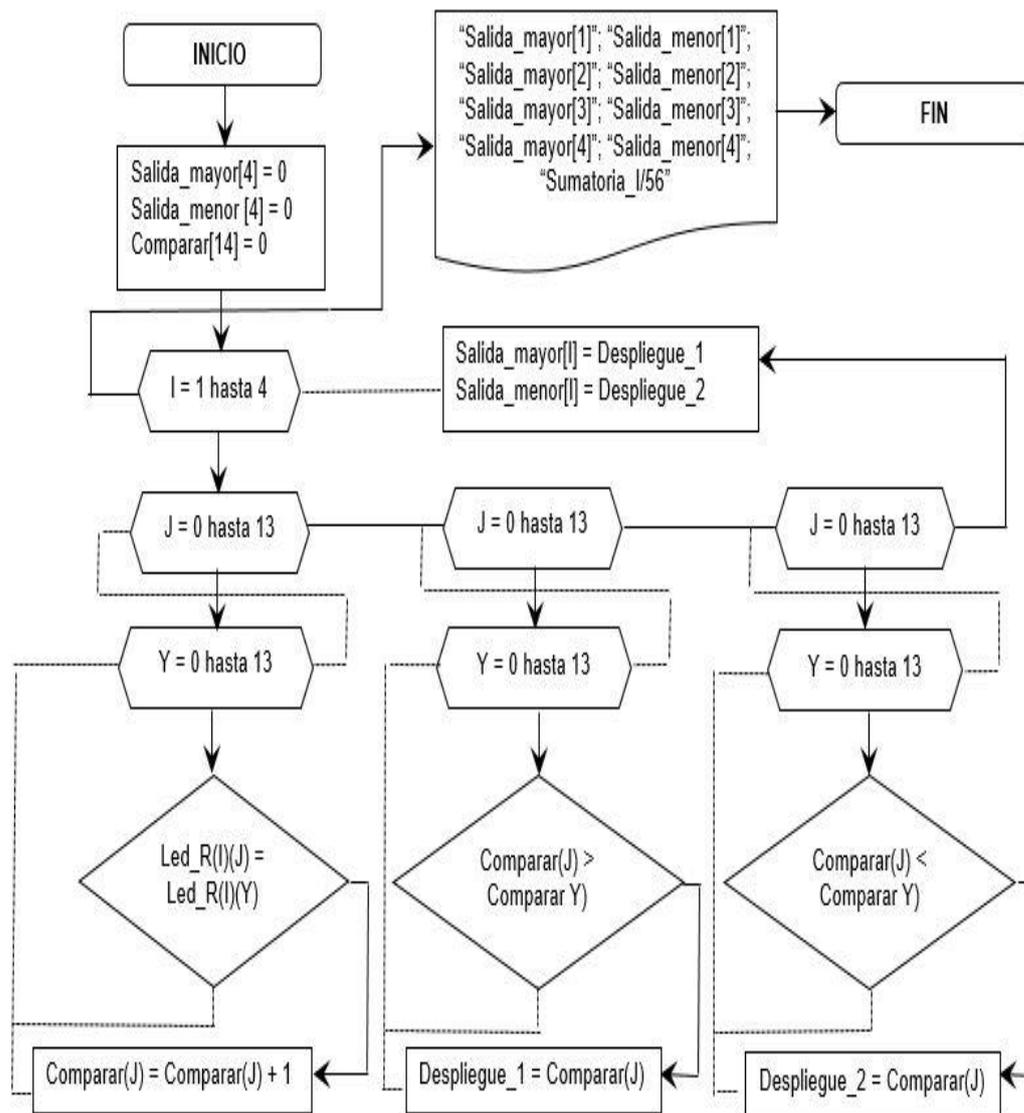


Figura 27. Procesamiento de muestras del led infrarrojo.

Como se mencionó en el capítulo II se necesita una etapa de control que se encargue de encender alternadamente los leds rojo e infrarrojo del sensor, para que puedan ser medidos independientemente, cabe recalcar que no es posible poder prender los dos leds al mismo tiempo. El integrado L293D, es el encargado del encendido alternado de los leds.

Se usan los pines 15 y 10 del integrado L293D que corresponden a las entradas de los circuitos push-pull 3 y 4, estas entradas están conectadas a las salidas D3 y D5 del Atmega 164 P, el pin 9 del integrado L293 D corresponde al enable el cual siempre tendrá un nivel alto (1), con estos parámetros se obtiene un resultado a la salida de los circuitos push-pull, en este caso como se usa los pines 15 y 10 se obtendrá cada 4 segundos, un nivel alto (1) o un nivel bajo (0) en los pines 11 y 14 que si se observa en la Figura 17, estos pines corresponden a las salidas de los circuitos push-pull 3 y 4 del integrado L293D. El resultado que se obtiene de los pines 11 y 14 del integrado L293D serán enviados a los pines de polarización del sensor para de esta manera saber cuál led encender, en la Figura 28, se representa el diagrama de conexión de la etapa de control.

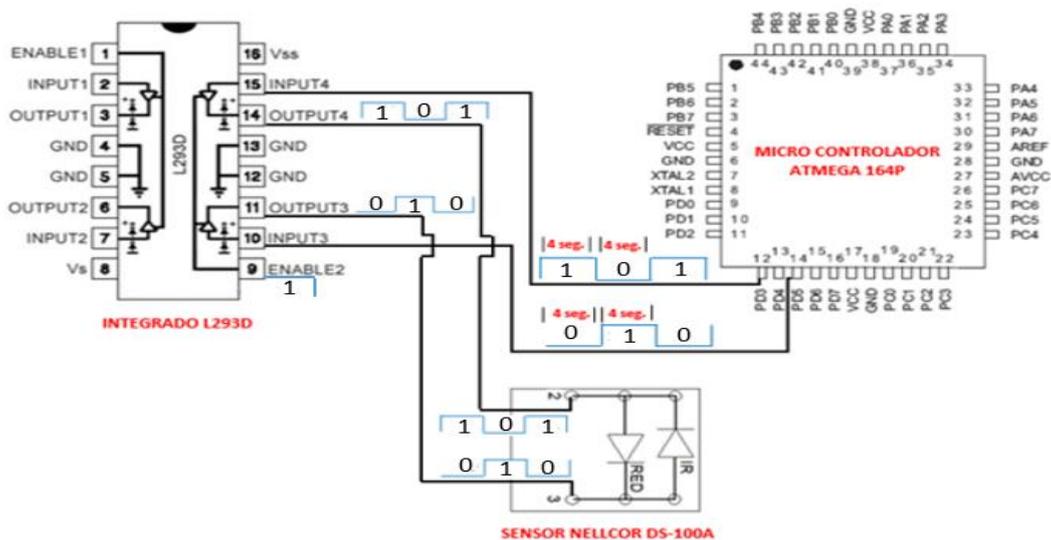


Figura 28. Etapa de control del sistema de monitoreo de control.

La tabla 20 representa el funcionamiento y control de encendido de los leds.

Tabla 20.

Control de encendido de los leds

Enable (PIN 9)	Entrada circuito 3 (PIN 10)	Entrada circuito 4 (PIN 15)	Salida circuito 3 (PIN 11)	Salida circuito 4 (PIN 14)	Función
1	1	0	1	0	Encendido led infrarrojo
1	0	1	0	1	Encendido led rojo

Tomado de (Tecnimicro, s.f.).

Adicional el integrado L293D a más de encender alternadamente los leds también sirve como protección del Atmega164P, ya que una de las características del integrado L293D es que puede controlar corrientes de hasta 600 mA, mientras que los pines del Atmega164P operan con una corriente de 40 mA, es decir si existe alguna falla, o alguna sobrecarga en el circuito, en el peor de los casos se dañaría el integrado L293 D, pero se procedería a su reemplazo sin ningún inconveniente, pero si no está presente el driver y existe una sobrecarga, se dañaría el Atmega164P.

3.4. Etapa de comunicación.

Una vez que la señal ha sido procesada es decir ha pasado por la etapa de amplificación diferencial, por el ADC y por la etapa de control, se tiene como resultado un numero de 10 bits que representa la medida analógica de la variación de los picos de voltaje provenientes de los leds rojo e infrarrojo, este número binario se guarda en los siguientes registros:

- **ADCH**

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
------	------	------	------	------	------	------	------

- **ADCL**

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
------	------	------	------	------	------	------	------

Cada registro guarda 8 bits, en el registro ADCH se almacenan los bits más significativos y en el ADCL se almacenan los registros menos significativos. (Micro controladores, s.f.).

Estas tramas de bits deben ser transferidos de alguna manera a un dispositivo móvil, para ello se implementa el módulo Bluetooth HC – 06 que mediante una comunicación serial con las salidas seriales D0 y D1 que corresponde a los pines (9 y 10) del microcontrolador Atmega 164 P, recibe la trama de bits.

La comunicación Bluetooth es una especificación tecnológica para redes inalámbricas y permite la transmisión de voz y datos entre distintos dispositivos mediante una radiofrecuencia segura (2,4 GHz). La comunicación Bluetooth es el medio que permite la sincronización del prototipo con el aplicativo Android, para esto se debe configurar una red de área personal siguiendo los siguientes pasos:

1. Conectar el Sensor Nellcor ds – 100A al prototipo.
2. Encender el prototipo.
3. Ingresar a la configuración Bluetooth del dispositivo móvil y buscar la conexión Bluetooth con el nombre de “OXIMETRO”.
4. Conectarse a “OXIMETRO” con la clave 5642.
5. Iniciar el aplicativo Android.

Realizado los pasos anteriores se comienza a transferir los datos, del Bluetooth al dispositivo móvil.

La Figura 29 representa el diagrama circuital del prototipo del sistema de monitoreo y control del pulso cardíaco y saturación de oxígeno en la sangre.

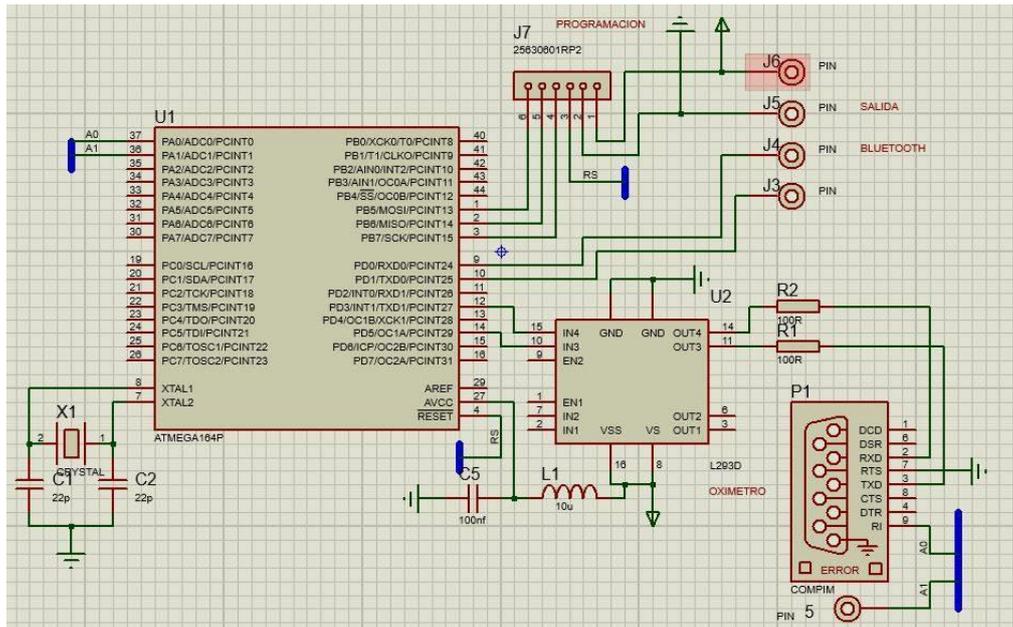


Figura 29. Diagrama circuital del prototipo del sistema de monitoreo y control del pulso cardíaco y saturación de oxígeno en la sangre.

3.5. Fuente de alimentación.

Como se mencionó en el capítulo II, se necesita de un voltaje de 5 V, para la alimentación del Bluetooth, y correcto funcionamiento del sistema de monitoreo, es por eso que 2 baterías de 3,5 V están conectadas en serie y con la ayuda de un regulador L7805CV, se mantiene constante el voltaje constante a 5 V, la estructura de la etapa de alimentación se representa en la Figura 30.

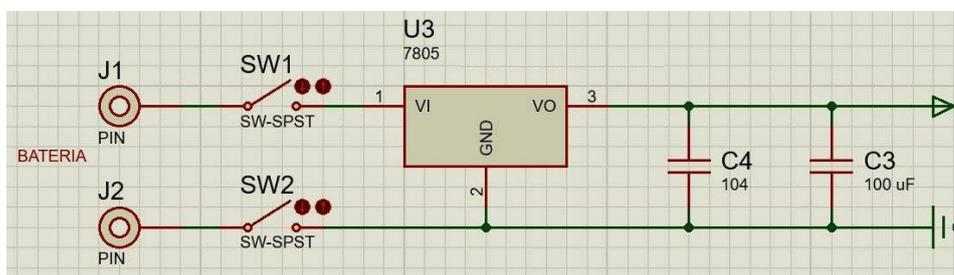


Figura 30. Diagrama circuital de la fuente de alimentación.

3.5. SQL SERVER motor de base de datos.

Todos los registros, datos personales tanto del paciente como el médico serán almacenados en una base de datos denominada “BDSISTEMA” para que luego estos datos pueden ser consultados y modificados si es necesario, la base de datos está distribuida mediante 4 tablas:

- **Doctores:** Almacena los datos personales del Médico tratante además permite el inicio de sesión al sistema.
- **Ingreso_Paciente:** En esta tabla el Médico ingresa los datos personales del paciente.
- **Control_Paciente:** Esta tabla principalmente almacena la información de los valores de la saturación de oxígeno en la sangre y el pulso cardíaco.
- **eMail:** Esta tabla está destinada para enviar los correos pendientes, trabaja conjuntamente con una computadora que hace de servidor.

El diseño de la base de datos “BDSISTEMA” se representa en la Figura 31.

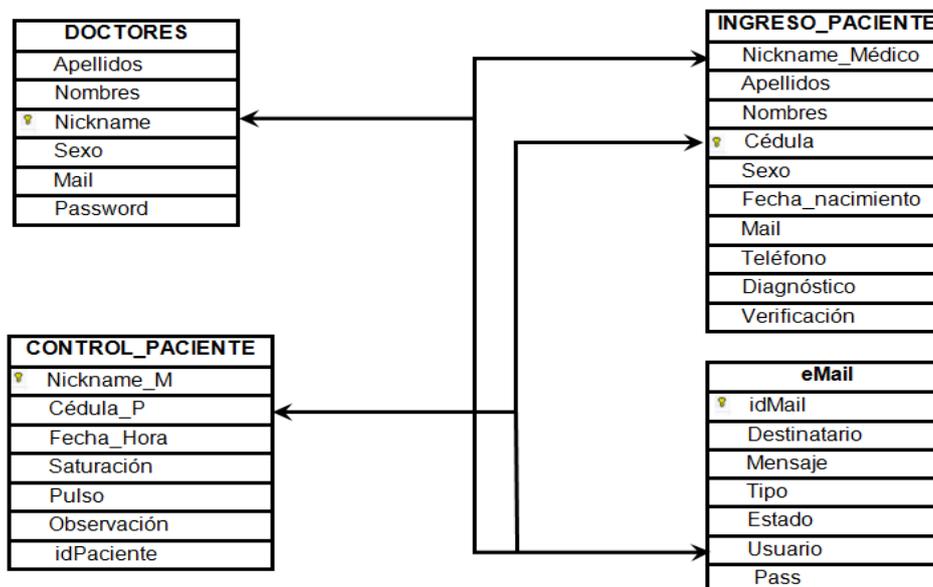


Figura 31. Diseño e la base de datos “BDSISTEMA”

El gráfico  representa a la llave principal de cada tabla la cual permite poder relacionar, los campos de las tablas entre sí, para de esta manera obtener los datos correctos y correspondientes a lo solicitado por el Médico o paciente. Finalmente, cuando se tenga el diseño concluido de la base de datos “BDSISTEMA” esta será almacenada en el Hosting somee.com, con el objetivo de que puede ser accesible desde cualquier dispositivo conectado a internet.

3.6. Diseño de la página web.

El software destinado para la comparación y gestión de los datos será implementado mediante el lenguaje de programación C#, el cual constará de cinco formularios, una página maestra y un Webservice. Adicional se requiere un servidor local que conjuntamente con un programa adicional implementado en C#, se logrará el envío de los mensajes de bienvenida al médico, al paciente, cambio de contraseña, y recordatorio de contraseña, por correo electrónico.

- **Formulario 1: Inicio**

Este formulario será el que se visualice cuando el Médico ingrese al software, la Figura 32 representa el diseño del formulario1.



Figura 32. Formulario 1

En este formulario si el Médico ya se ha registrado anteriormente, deberá ingresar su nombre de usuario y contraseña para ingresar al sistema o pasar al formulario 2, si ingresa un nombre de usuario o contraseña incorrecta o si deja los espacios en blanco y desea ingresar se desplegará los mensajes de error correspondientes.



Figura 33. Mensajes de error del formulario 1

Si es la primera vez que el Médico ingresa al software, deberá registrarse para que pueda tener acceso, dando el clic en la opción “Registrarse” que aparece en la Figura 32, ese link desplegará la siguiente ventana que se representa en la Figura 34.

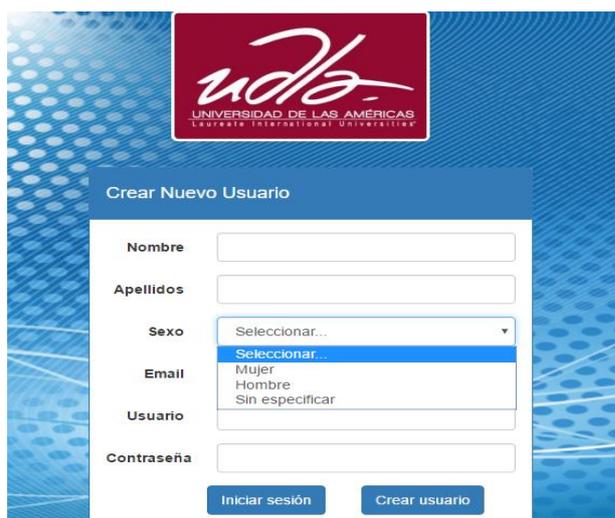


Figura 34. Registro y creación de usuario.

En esta ventana el Médico ingresará sus datos personales, se desplegará un mensaje de error en los siguientes casos:

- Si el médico ingresa un nombre de usuario ya existente o un correo electrónico ya asociado a alguna cuenta.
- Si el médico ingresa un correo de forma errónea o deja algún espacio en blanco.

En la Figura 35 se representan los mensajes de error que serán desplegados en el registro y creación de usuarios.



Figura 35. Mensajes de error del formulario 1

Pero si los datos son correctos se desplegará la advertencia representada en la Figura 36, comunicándole que el usuario fue creado, y que se le envió un mensaje de bienvenida a su correo personal, recordándole el usuario y contraseña, inmediatamente los datos ingresados serán almacenados en la tabla Doctores de la base de datos BDSISTEMA.

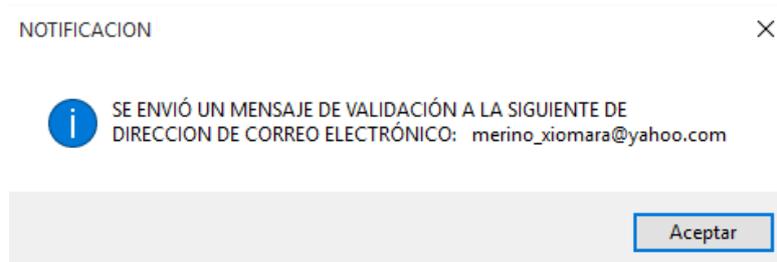


Figura 36. Mensaje de creación de usuario



Figura 37. Formato de mensaje enviado al mail personal

Cabe recalcar que el formato de los mensajes enviados al correo personal, dependerá de la opción que se escoja en la opción sexo del registro y creación de usuarios representados en la Figura 34.

- **Diagrama de flujo.**

El diagrama de flujo del formulario1, se representa en la Figura 37.

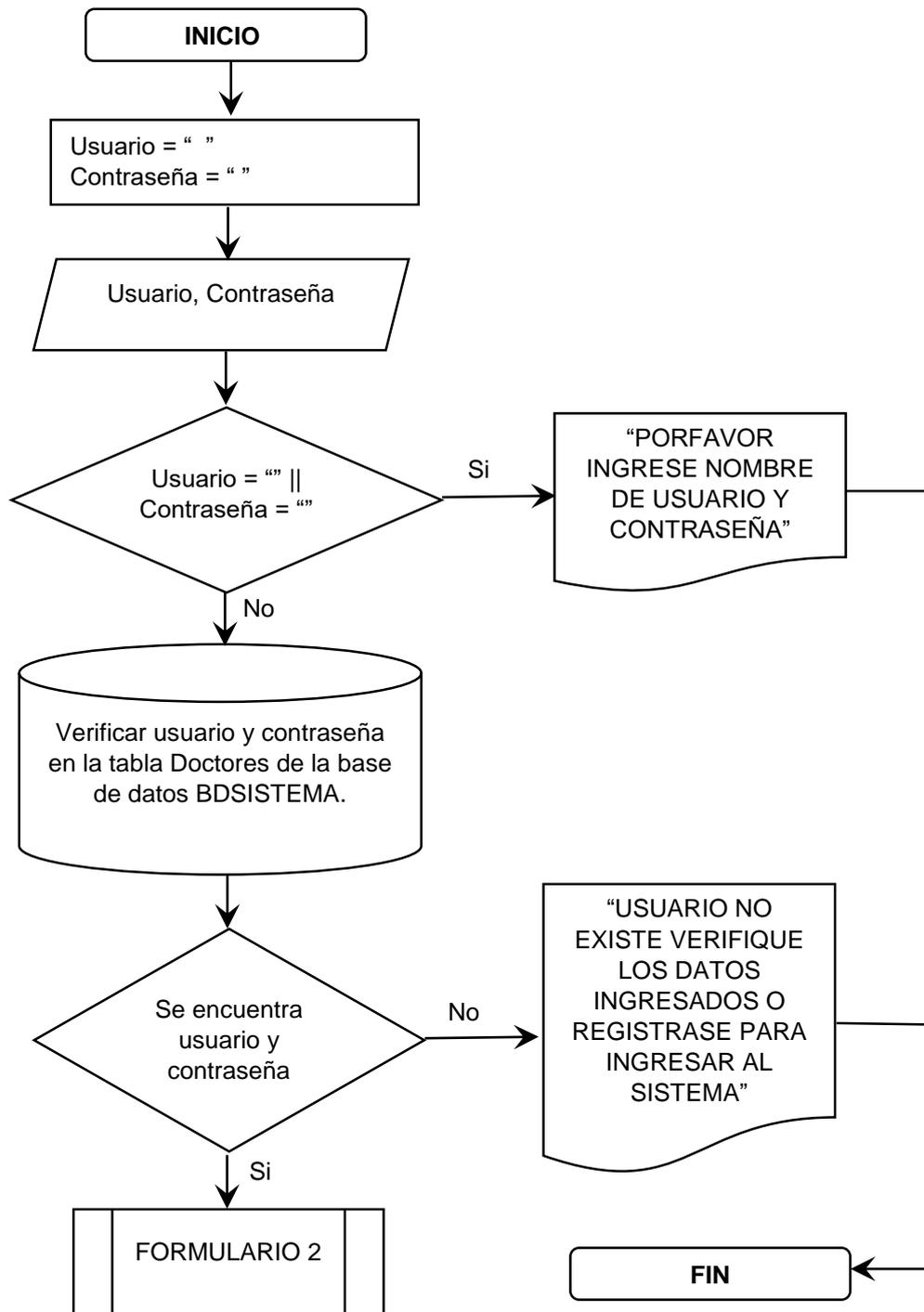
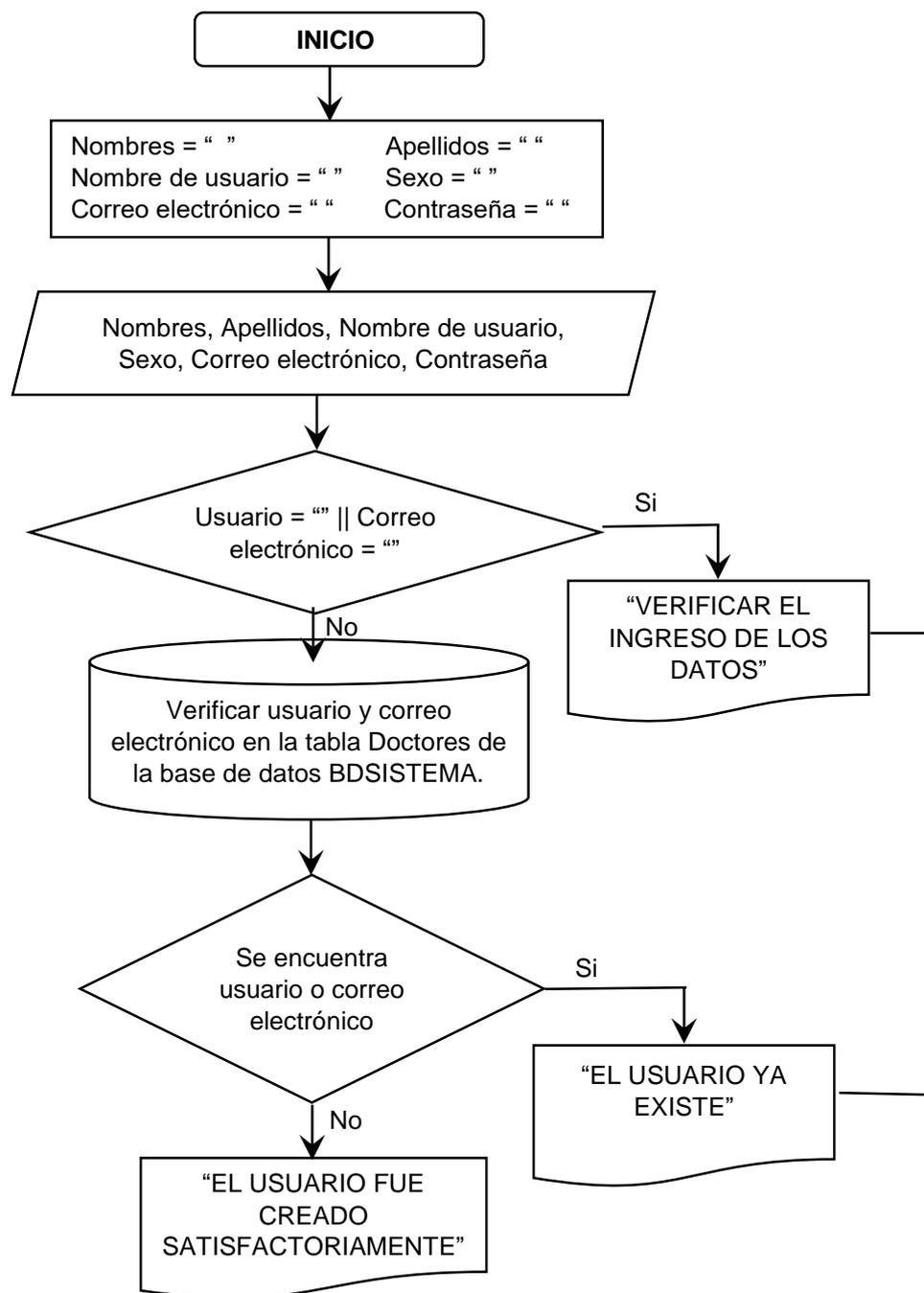


Figura 38. Diagrama de flujo del formulario 1

- Diagrama de flujo del registro y creación de usuarios.



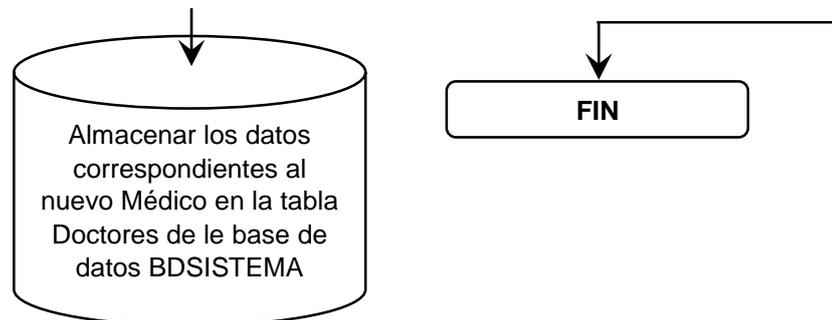


Figura 39. Diagrama de flujo del registro y creación de usuario.

- **Formulario 2:** Presentación

Una vez que el Médico ha sido registrado y se ha iniciado sesión correctamente, se despliega el formulario 2 que le da la bienvenida con su acrónimo y su nombre.

El acrónimo del Médico en el formulario 2 dependerá de la opción sexo con el que se registre el Médico en la opción registro y creación de usuario de la Figura 34, las Figuras 40, 41 y 42 muestran los resultados que se podrán obtener en el formulario 2.



Figura 40. Formulario 2 si se escoge en sexo la opción Mujer.



Figura 41. Formulario 2 si se escoge en sexo la opción Hombre.



Figura 42. Formulario 2 si se escoge en el sexo "Sin especificar".

Como se puede observar en la parte superior del formulario hay 2 botones uno con la opción "Ingresar paciente" y el otro con la opción "Consultar paciente", dependiendo de las necesidades del Médico serán escogidos y se desplegará a otro formulario.

- **Diagrama de flujo del formulario 2.**

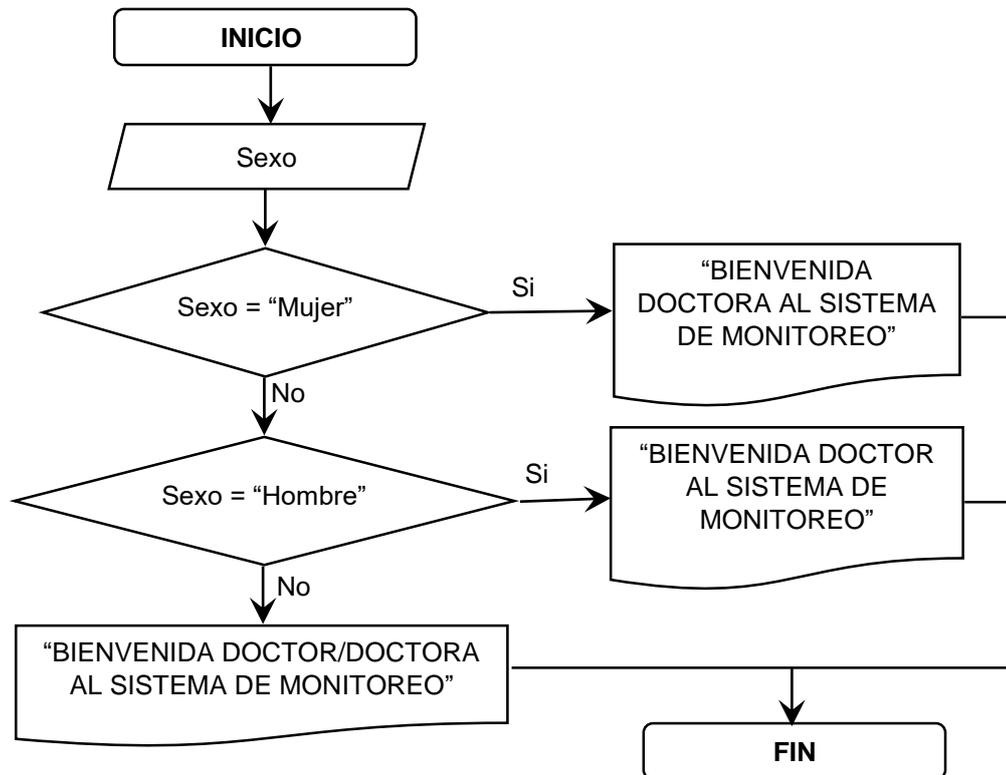
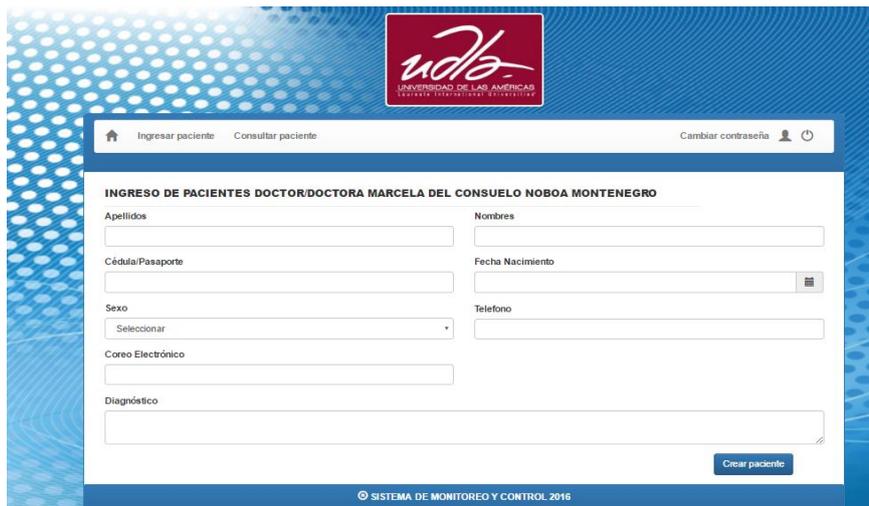


Figura 43. Diagrama de flujo del formulario 2

- **Formulario 3: Ingresar paciente**

El formulario 3 será destinado para el ingreso y registro de los pacientes del médico y su despliegue dependerá de que botón se presione en el formulario 2 en este caso si se presiona el botón con el mensaje "Ingresar paciente" se desplegará el formulario 3 que está representada en la Figura 44.



The image shows a web-based patient registration form. At the top, there is a navigation bar with links for 'Ingresar paciente', 'Consultar paciente', and 'Cambiar contraseña'. The form title is 'INGRESO DE PACIENTES DOCTOR/DOCTORA MARCELA DEL CONSUELO NOBOA MONTENEGRO'. The form fields include: 'Apellidos', 'Nombres', 'Cédula/Pasaporte', 'Fecha Nacimiento', 'Sexo' (with a dropdown menu), 'Telefono', 'Correo Electrónico', and 'Diagnóstico'. A 'Clear paciente' button is located at the bottom right of the form. The footer of the page reads '© SISTEMA DE MONITOREO Y CONTROL 2016'.

Figura 44. Presentación del formulario 3

En este formulario se tomará el dato ingresado en el campo cédula/pasaporte para la verificación de que el paciente está registrado o no, es decir si el software detecta que el dato ingresado en cédula/pasaporte ya existe en la base de datos BDSISTEMA, se desplegará un mensaje con una advertencia, de igual manera si se deja algún espacio en blanco o si el correo está mal escrito se desplegará un mensaje de error, los mensajes de error desplegados en este formulario están representados en las Figuras 45 y 46 respectivamente.

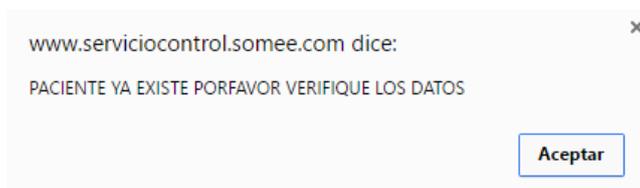


Figura 45. Mensaje de error si el paciente ya existe

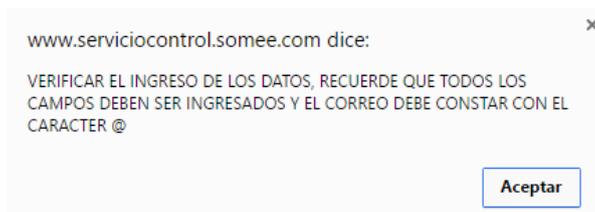


Figura 46. Mensaje de error para espacios en blanco o correo mal escrito

Si el número de cédula o el pasaporte no se encuentra registrados y no existe espacios en blanco, el sistema permitirá la creación del paciente y desplegará un mensaje confirmando que el usuario fue creado exitosamente, el mensaje que será desplegado está representado en la Figura 47, e inmediatamente se enviará un mensaje de bienvenida al correo personal del paciente con el programa adjunto para que se descargue en su teléfono celular.

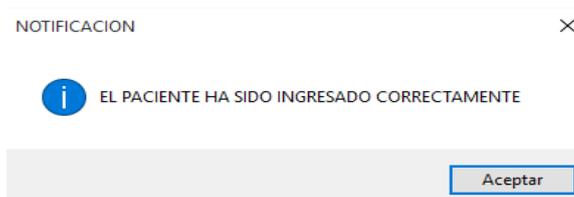


Figura 47. Mensaje de creación de paciente.



Figura 48. Formato de mensaje enviado al mail personal del paciente

- Diagrama de flujo del formulario 3

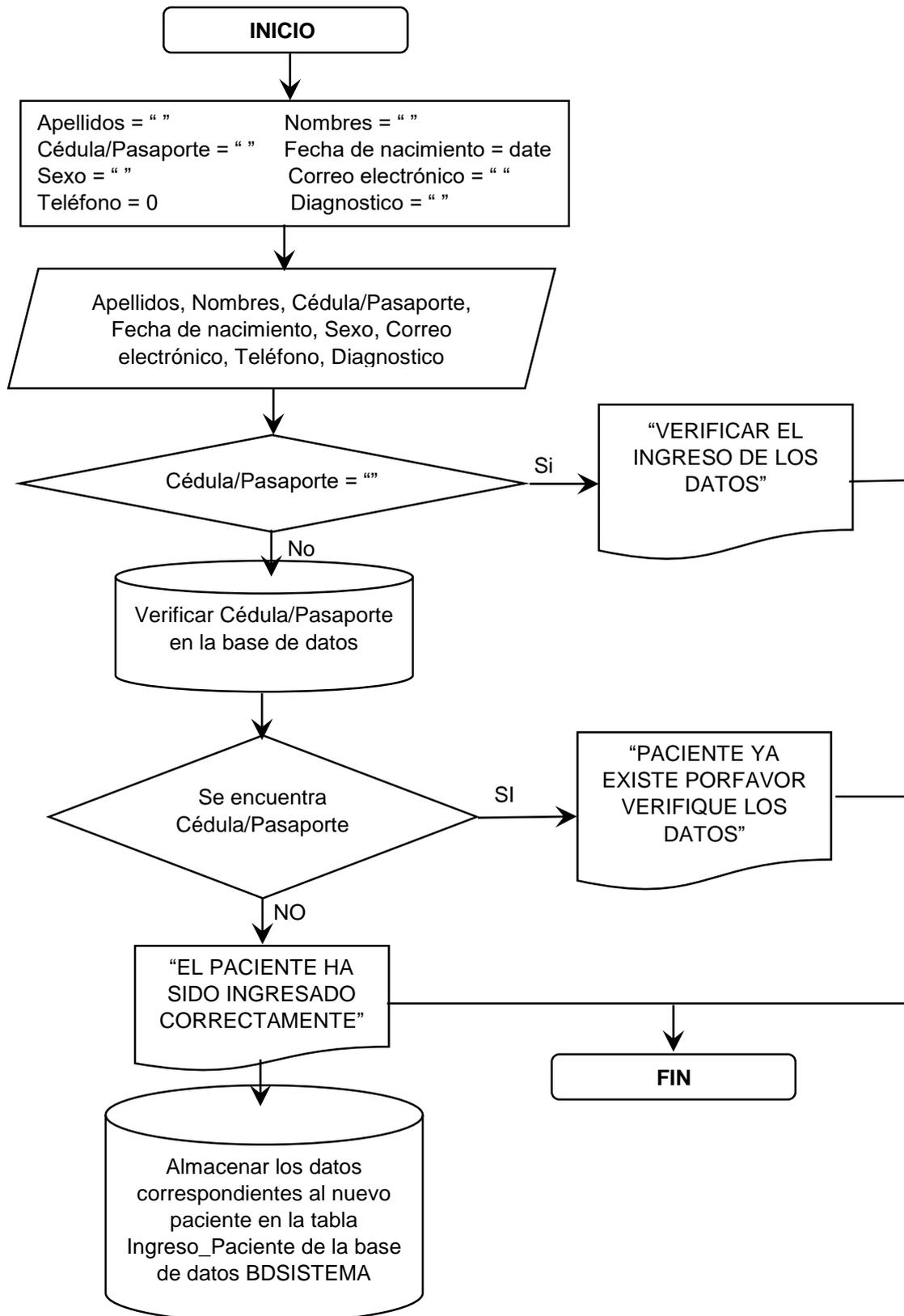


Figura 49. Diagrama de flujo del formulario 3.

- **Formulario 4:** Controlar paciente

El formulario 4 será destinado para un control continuo de los pacientes por parte del médico tratante y su despliegue dependerá de que botón se presione en el formulario 2 en este caso si se presiona el botón con el mensaje “Consultar paciente” se desplegará el formulario 4 que está representado en la Figura 50.

Nickname_M	Cédula_P	Paciente	Fecha_Hora	Saturación	Pulso	Observación	idPaciente
Select david	1717171712	vazquez juan	18/11/2016 7:15:36	71.0 SPO2	73.0 PR	Saturacion baja y pulso cardiaco inestable	140

© SISTEMA DE MONITOREO Y CONTROL 2016

Figura 50. Presentación del formulario 4.

Para que el Médico pueda tener un control adecuado de los pacientes, el formulario 4 contiene un GridView el cual es un control que permite visualizar datos en una tabla donde cada columna representa un campo y una fila representa un registro.

En el formulario 4 el GridView obtendrá los registros de la tabla Control_Paciente de la base de datos BDSISTEMA, correspondientes a los eventos ocurridos por una baja saturación de oxígeno en la sangre de sus pacientes. Adicional el formulario 4 tiene una opción de búsqueda avanzada ya sea por nombre o cédula del paciente con el objetivo de facilitar la búsqueda si el Médico desea chequear un paciente específico.

Cabe recalcar que el Médico únicamente podrá observar y modificar los registros de sus pacientes, es importante tener en cuenta esto, debido a que la tabla Control_Paciente de la base de datos BDSISTEMA tendrá todos los eventos producidos por una baja saturación de oxígeno en la sangre de los pacientes de todos los Médicos registrados en el sistema, pero el médico solo tendrá acceso al registro de sus pacientes.

- **Diagrama de flujo del formulario 4.**

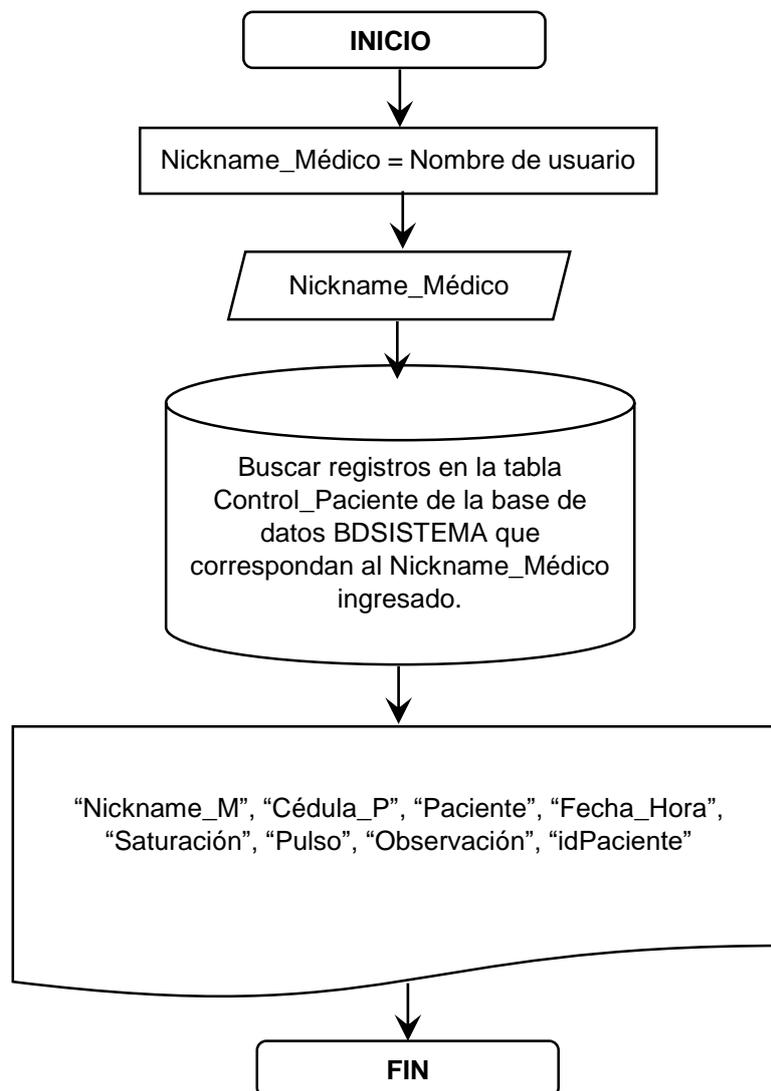


Figura 51. Diagrama de flujo del formulario 4

- **Página maestra:** Principal.

La función principal de las paginas maestras es permitir al programador crear un único diseño que será compartido para 2 o más formularios, ahorrando de esta manera tiempo en el diseño de un nuevo formulario.

Si se observa nuevamente las Figura 52, 53 y 54 que corresponden, al formulario 2, formulario 3 y formulario 4, se puede ver, que el diseño de los formularios tiene la misma estructura esto se debe a que comparten una misma página maestra denominada Principal, en la Figura 52 se aprecia el diseño de la página maestra Principal.

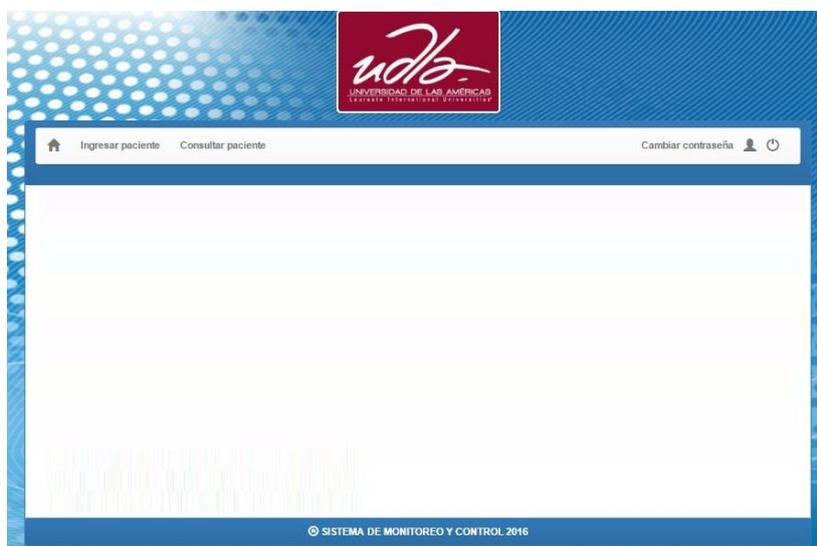


Figura 52. Diseño de la página maestra

Adicional el software contará con dos opciones adicionales que son, recordar contraseña y cambiar contraseña.

- **Recordar contraseña:** Esta opción fue adicionada con el objetivo de que si el Médico, no recuerda su contraseña puede recibirla por correo electrónico, ya sea ingresando su usuario o correo electrónico al cual tiene asociado en el sistema, la Figura 53 representa el diseño de la opción recuperar contraseña.

UNIVERSIDAD DE LAS AMÉRICAS
Laureate International Universities

Recuperar Datos de Usuario

Usuario

Email

Iniciar sesión Recuperar contraseña

Figura 53. Diseño de la opción recordar contraseña

Como en los demás diseños anteriores se debe ingresar datos que consten en el sistema, si los datos ingresados no están registrados, los espacios están en blanco o el correo está mal ingresado y se solicita recuperar la contraseña, saldrán los mensajes de advertencia correspondientes los cuales se muestran en las Figuras 54 y 55.

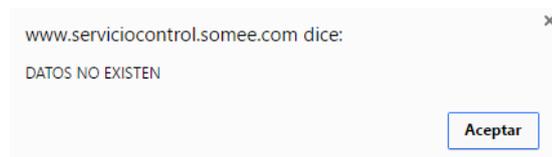


Figura 54. Mensaje de error si los datos no están registrados

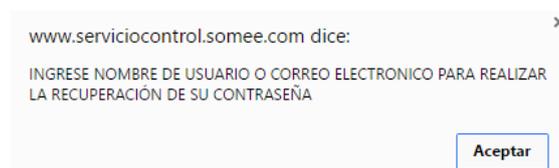


Figura 55. Mensaje de error si los dos espacios están en blanco

Si los datos ingresados se encuentran registrados en el sistema y por lo menos una opción no está en banco, se desplegará un mensaje diciendo que se envió el usuario y contraseña a la dirección de correo electrónico asociada a la cuenta.

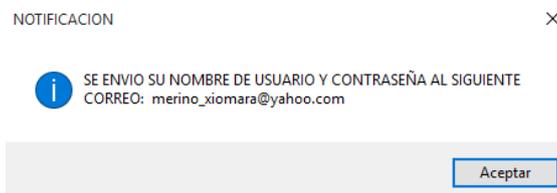


Figura 56. Mensaje de aviso que la contraseña fue recuperada



Figura 57. Formato de mensaje enviado al mail personal

Cabe recalcar que de igual manera el formato del mensaje enviado por correo electrónico cambiara dependiendo de la opción escogida en el sexo de la Figura 34.

- **Diagrama de flujo.**

El diagrama de flujo de la opción recuperar contraseña, se representa en la Figura.

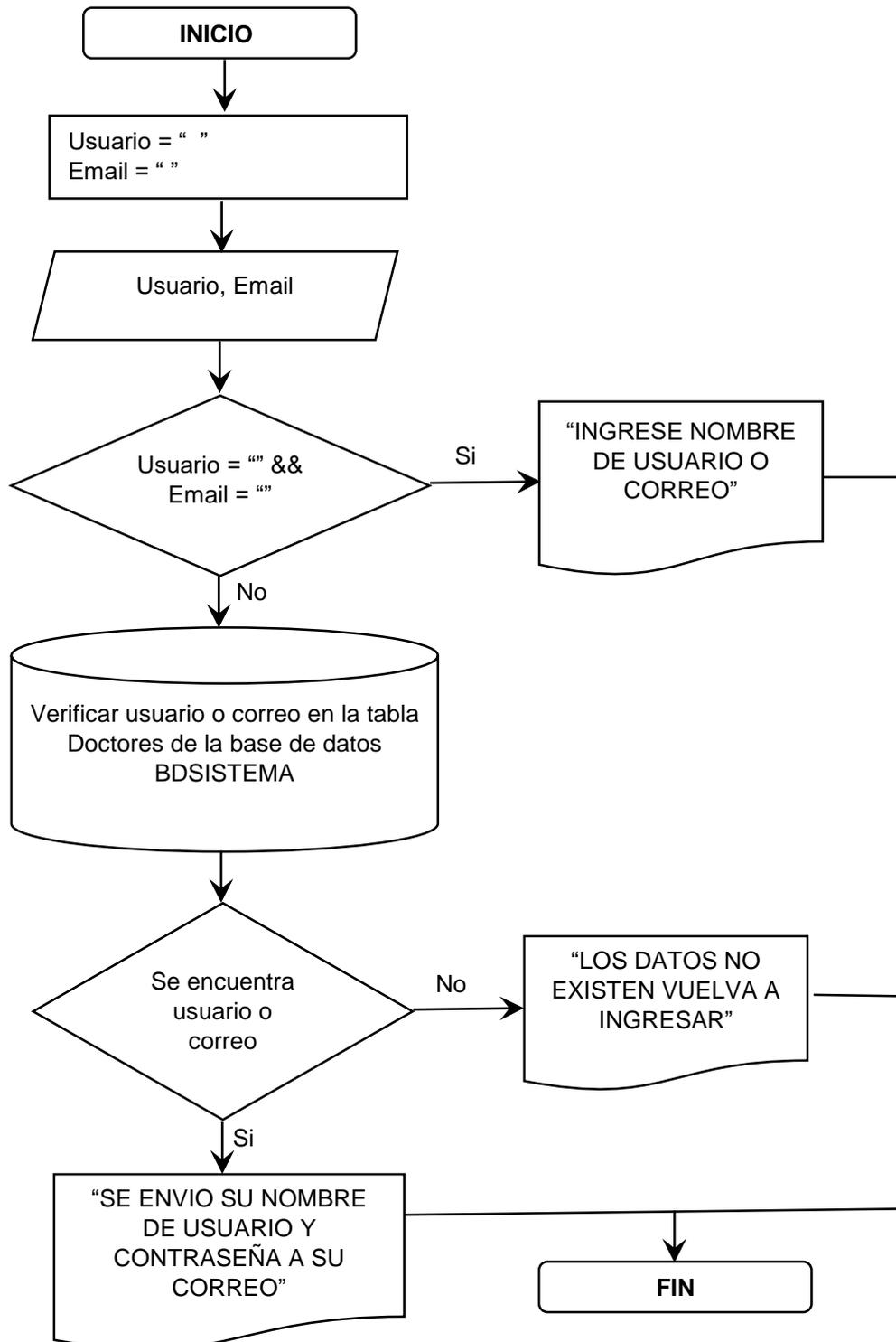


Figura 58. Diagrama de flujo de la opción recuperar contraseña.

- **Cambiar contraseña:** El Médico tendrá la opción de cambiar la contraseña si no está a gusto con la anterior, la opción cambio de contraseña viene a ser el formulario 5, para realizar dicho requerimiento se debe ingresar 2 veces la contraseña nueva.



The image shows a web interface for user recovery. At the top, there is a logo for 'UNIVERSIDAD DE LAS AMÉRICAS' with the tagline 'Laureate International Universities'. Below the logo is a blue header with the text 'Recuperar Datos de Usuario'. The main form area contains two input fields: 'Usuario' and 'Email'. Below these fields are two buttons: 'Iniciar sesión' and 'Recuperar contraseña'. The background of the page features a blue pattern of dots and lines.

Figura 59. Diseño de la opción cambiar contraseña

Al igual que los controles anteriores, si el Médico deja los espacios en blanco, los datos ingresados no coinciden, o la contraseña nueva ya existe, se desplegarán los diversos mensajes de advertencias los cuales están representados en las Figuras 60, 61 y 62.

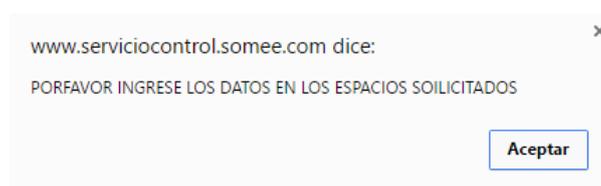


Figura 60. Mensaje de error si los dos espacios están en blanco

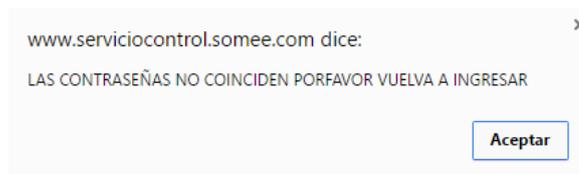


Figura 61. Mensaje de error si las contraseñas no coinciden

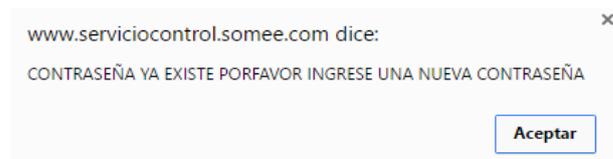


Figura 62. Mensaje de error si la contraseña ya existe

Si los datos ingresados son los correctos, se desplegará una notificación informando que se ha enviado un mensaje al correo electrónico dándole a conocer su nueva contraseña e inmediatamente será re direccionado a la ventana de login.

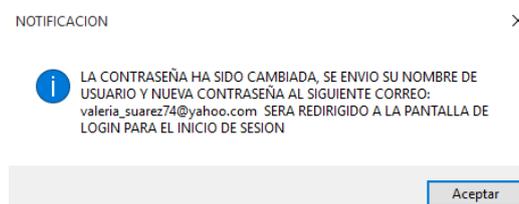


Figura 63. Mensaje de aviso que la contraseña ha sido reemplazada



Figura 64. Formato de mensaje enviado al mail personal

El mensaje enviado por correo electrónico cambiara dependiendo de la opción escogida en el sexo de la Figura 34.

- Diagrama de flujo

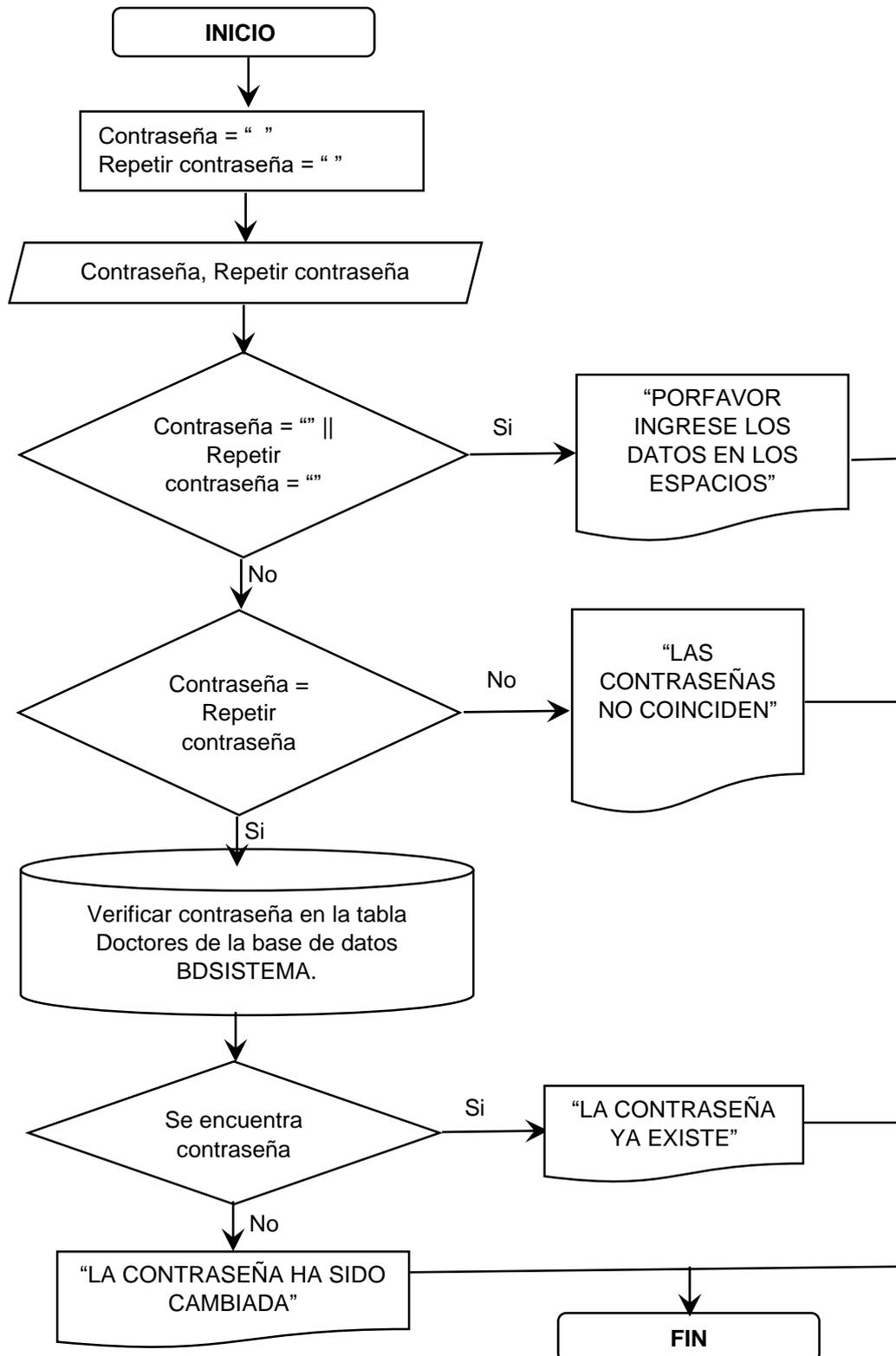


Figura 65. Diagrama de flujo de la opción cambiar contraseña.

- **WebService:** WebService.

El WebService corresponde a un conjunto de protocolos y estándares que son usados para intercambiar y transferir datos entre aplicaciones.

En el sistema de monitoreo y control, el WebService fue diseñado en C # asp.net, con el objetivo de permitir la transferencia de datos e interacción entre la base de datos BDSISTEMA y el aplicativo Android, en la Figura 63 se representa el funcionamiento del WebService.



Figura 66. Funcionamiento del WebService

Tomado de (Walter, 2014)

- **Servidor local.**

Para él envío de los mensajes de bienvenida tanto para el médico y paciente, y los mensajes de cambio de contraseña y recordatorio de contraseña, fue necesario, montar un servidor local para que los mensajes sean enviados a los correos personales.

Para este agregado se usa la opción Programador de tareas, para ingresar a esta opción se sigue el siguiente procedimiento: menú Inicio > Accesorios > Herramientas del sistema > Programador de tareas.

El programador de tareas básicamente ayuda a programar tareas que realizan acciones en un tiempo determinado, cuando se produce algún evento.

En el sistema de monitoreo el programador de tareas trabajará conjuntamente, con un programa diseñado en C#, dicho programa contendrá el formato del mensaje a ser enviado y a su vez se conectará con la tabla eMail de la base de datos BDSISTEMA, con el objetivo de verificar si se debe o no enviar el mensaje al correo electrónico y que formato de mensaje debe ser enviado, ya que como se explicó anteriormente el formato cambia dependiendo de la acción que se realice.

- Diagrama de flujo.

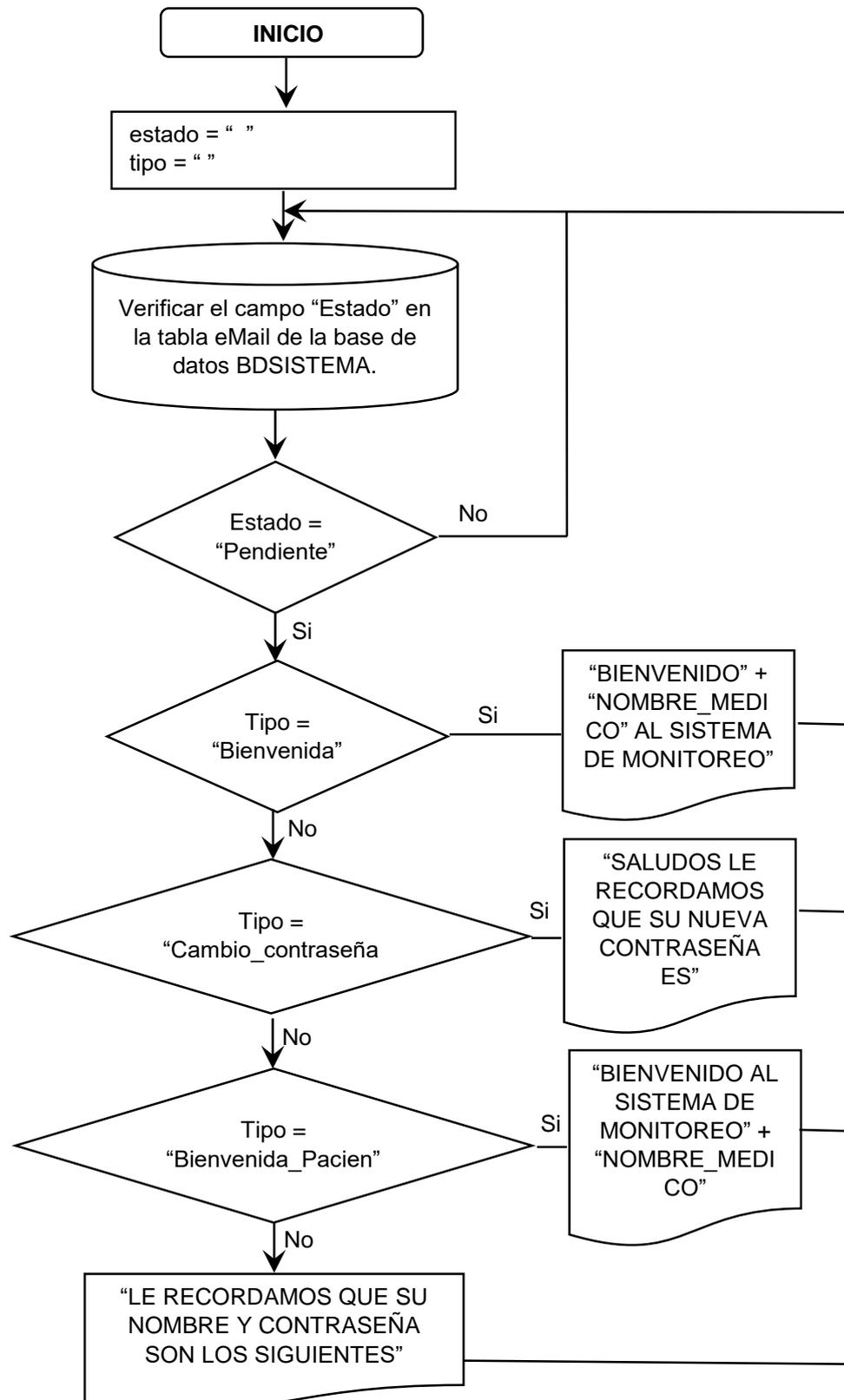


Figura 67. Diagrama de flujo para el envío de correos.

El proceso detallado en el diagrama de flujo para el envío de correos representado en la Figura 67 se repetirá cada minuto, es decir en el transcurso de un minuto el programador de tareas ejecutara en el servidor local el programa diseñado para el envío de los mensajes al correo electrónico.

3.7. Aplicativo en Android.

El aplicativo Android destinado para el gestiona miento y procesamiento de los valores correspondiente a la saturación de oxígeno en la sangre y pulso cardíaco obtenidos por el dispositivo electrónico será implementado mediante programación Android, usando la plataforma eclipse, el aplicativo constará de dos layouts.

- **Layout 1:** activity_login

El layout 1 será lo primero a visualizar en el teléfono móvil cuando el paciente ejecute el aplicativo Android, en la Figura 68 se representa el diseño del layout 1.



Figura 68. Diseño del layout 1.

En este layout si el paciente ya ha sido registrado en el software de comparación y gestión de datos por su médico tratante, deberá ingresar su cédula o pasaporte para ingresar al sistema o pasar al layout 2, si ingresa una cédula o pasaporte incorrecto o si deja los espacios en blanco se desplegará los mensajes de error correspondientes, las Figuras 69 y 70 representan los mensajes de error que se tendrán en el layout 1.



Figura 69. Mensaje de error si el campo para inicio de sesión esta en blanco



Figura 70. Mensaje de error a una clave de inicio de sesión errónea.

- Diagrama de flujo.

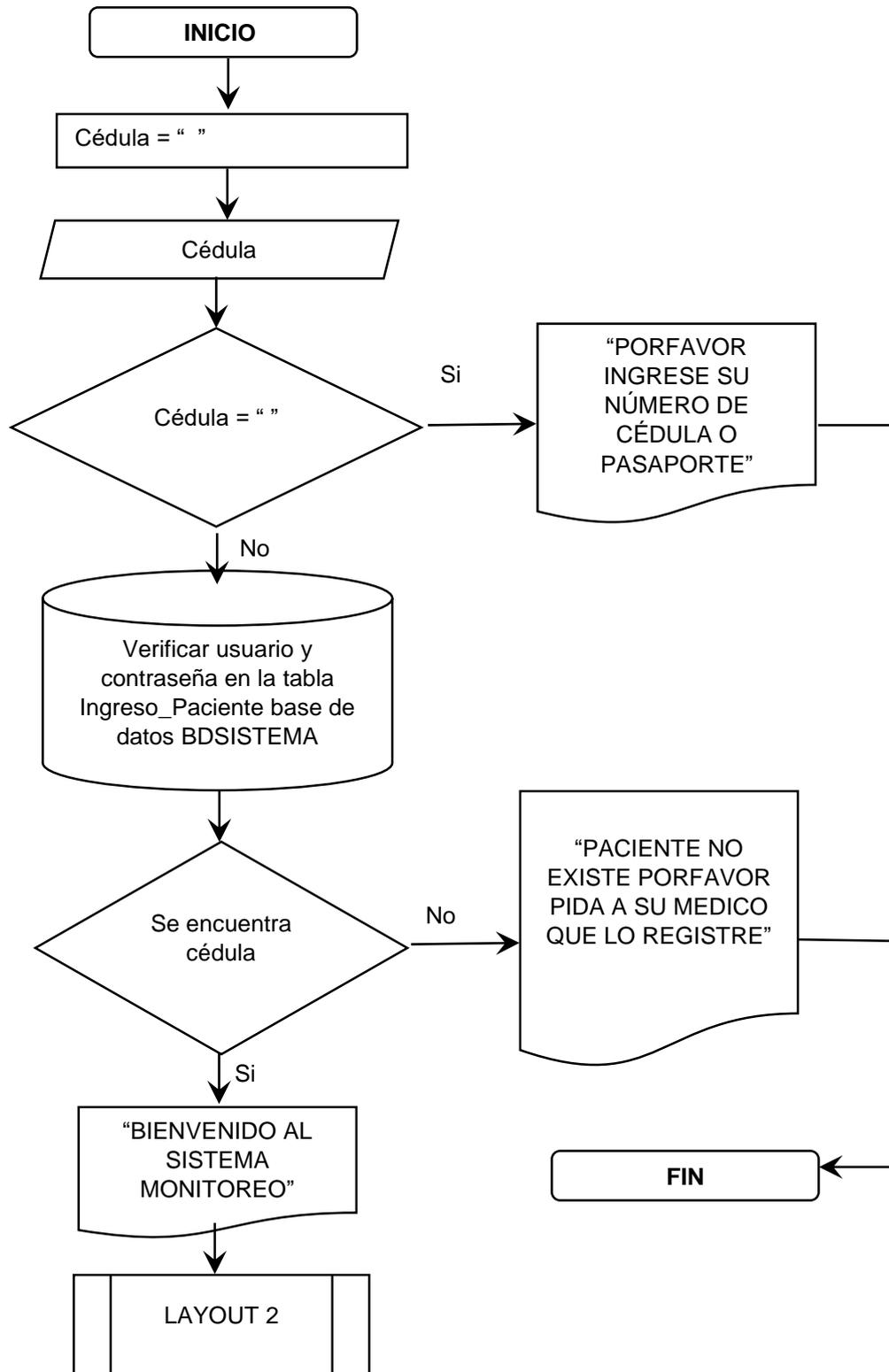


Figura 71. Diagrama de flujo correspondiente al layout 1.

- **Layout 2:** activity_sistema_monitoreo.

Una vez que el paciente ha iniciado sesión correctamente, se despliega el layout 2. En el layout 2 el sistema solicitará los permisos para activación del bluetooth y sincronización del aplicativo Android con el prototipo, para de esta manera empezar la transferencia y procesamiento de los datos entre el circuito electrónico y el teléfono celular.

- **Obtención del pulso cardíaco:**

El número de muestras de mayor valor que se obtiene de la Figura 23, significa el número de pulsos (# pulsos) que una persona ha obtenido en 8 segundos, y como el pulso cardíaco es el número de latidos que una persona tiene en un minuto se establece una regla de tres simple.

8 segundos \longrightarrow # pulsos

60 segundos \longrightarrow PR

Para la obtención del pulso cardíaco se establece la siguiente ecuación.

$$PR = \frac{60 * \# \text{ pulso}}{8 \text{ segundos}} \quad (\text{Ecuación 2})$$

- **Obtención de la saturación de oxígeno:**

Para la obtención de la saturación de oxígeno en la sangre, se obtiene un promedio de las muestras representadas en la Figura 26 y 27, y se establece la siguiente ecuación (Ronquillo, J. y Salgado, P, Diseños y construcción de un oxímetro de pulso portátil, 2012).

$$SP02 = \frac{PR}{PI+PR} * 100 \quad (\text{Ecuación 3})$$

}

PR = Promedio de muestras del led rojo

PI = Promedio de muestras del led infrarrojo

Cuando la transferencia de datos da inicio, el aplicativo Android, realizará una comparación de todos los datos que le llegan, y si existe alguna anomalía, inmediatamente se desplegará un mensaje de alerta en el celular, y los datos responsables de activar el mensaje serán almacenados en la tabla Control_Paciente de la base de datos BDSISTEMA, para que el médico tratante puede analizar los datos mediante el software de comparación y gestión de datos, las Figuras 72, 73, 74 y 75 representan los eventos que se presentaran en el layout 2.

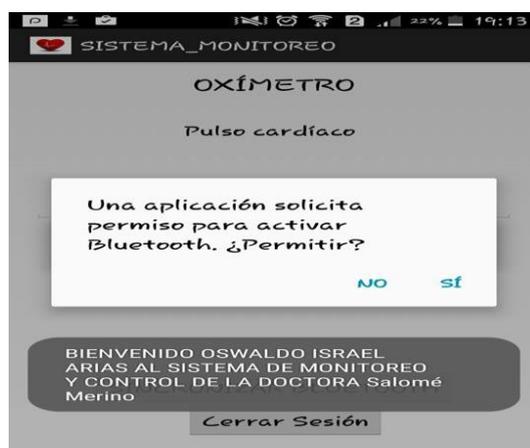


Figura 72. Solicitud de permiso para encender Bluetooth



Figura 73. Sincronización del aplicativo Android con el prototipo.



Figura 74. Transferencia de datos

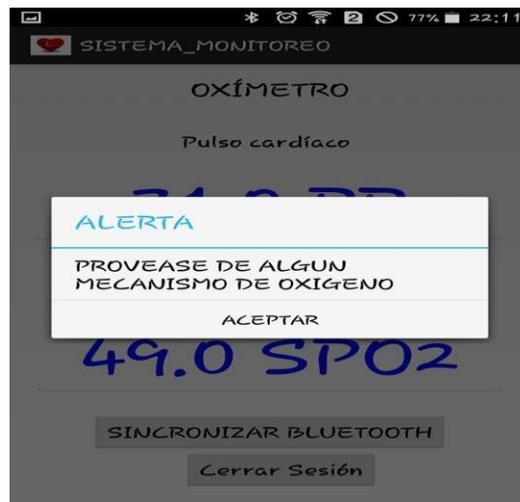


Figura 75. Mensaje de alerta

Es importante tomar en cuenta los siguientes aspectos para el desarrollo del aplicativo Android:

- Para el inicio de sesión y para el almacenamiento de los datos cuando la alarma sea activada se lo realiza mediante el Web Service que como se mencionó anteriormente es el encargado de establecer la transferencia de datos entre el aplicativo Android y la base de datos BDSISTEMA.

- El aplicativo Android, transfiere los registros a la base de datos BDSISTEMA a través del internet y también recibe datos del prototipo a través de bluetooth, es por ese motivo que para el funcionamiento correcto del aplicativo Android se debe realizar las configuraciones básicas para obtener los permisos de internet como bluetooth, esta configuración se la realiza en el AndroidManifest.xml, el cual es un archivo para aplicar configuraciones básicas de una app.

En la Figura 76 se puede apreciar el AndroidManifest.xml con sus respectivas configuraciones de internet y bluetooth.

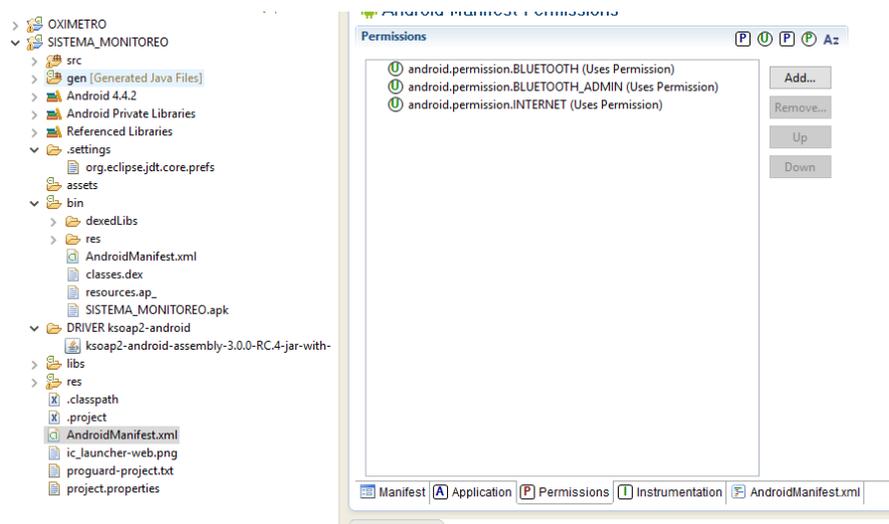
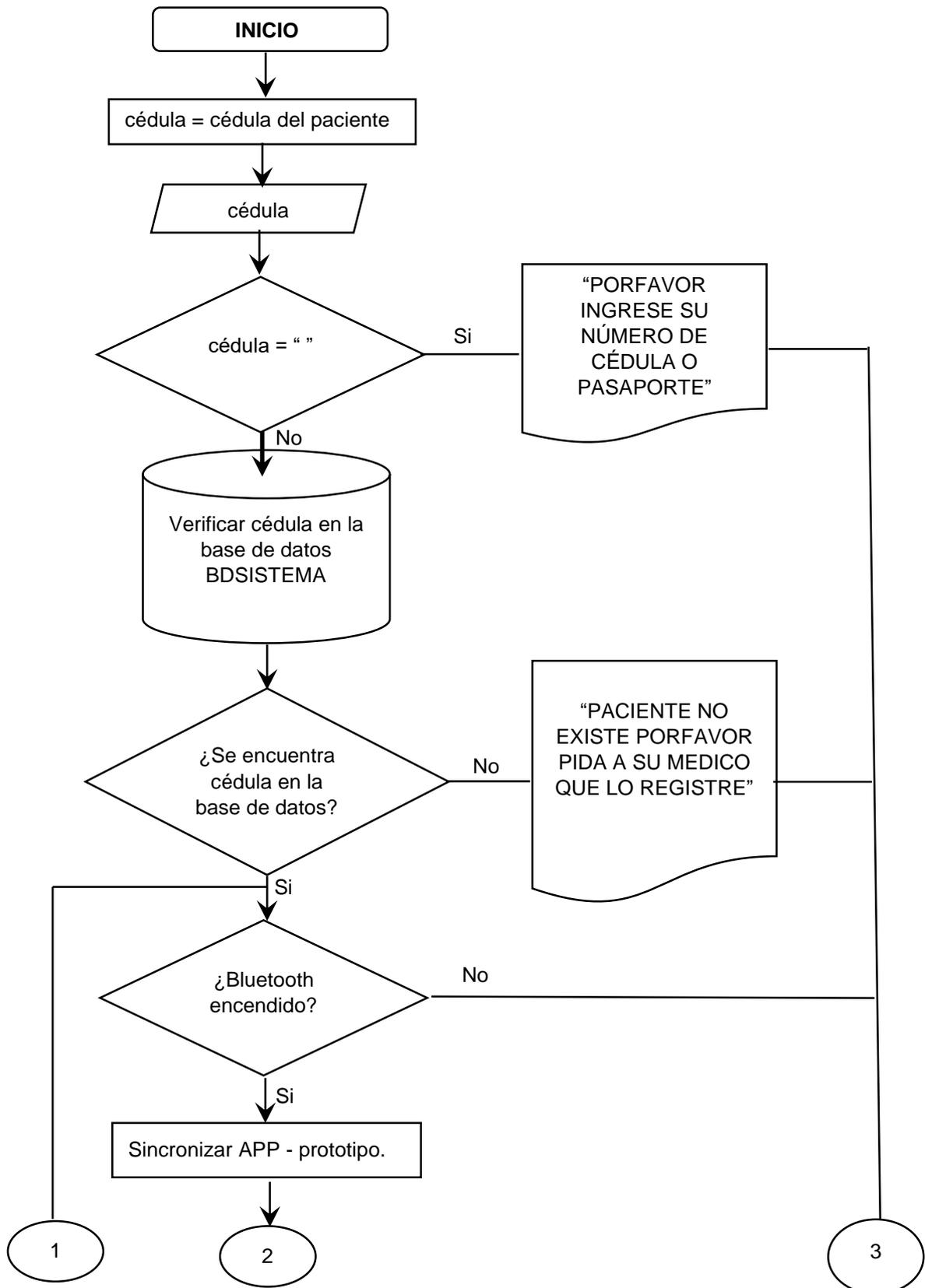


Figura 76. AndroidManifest.xml y su configuración básica para el funcionamiento del prototipo

- Diagrama de flujo



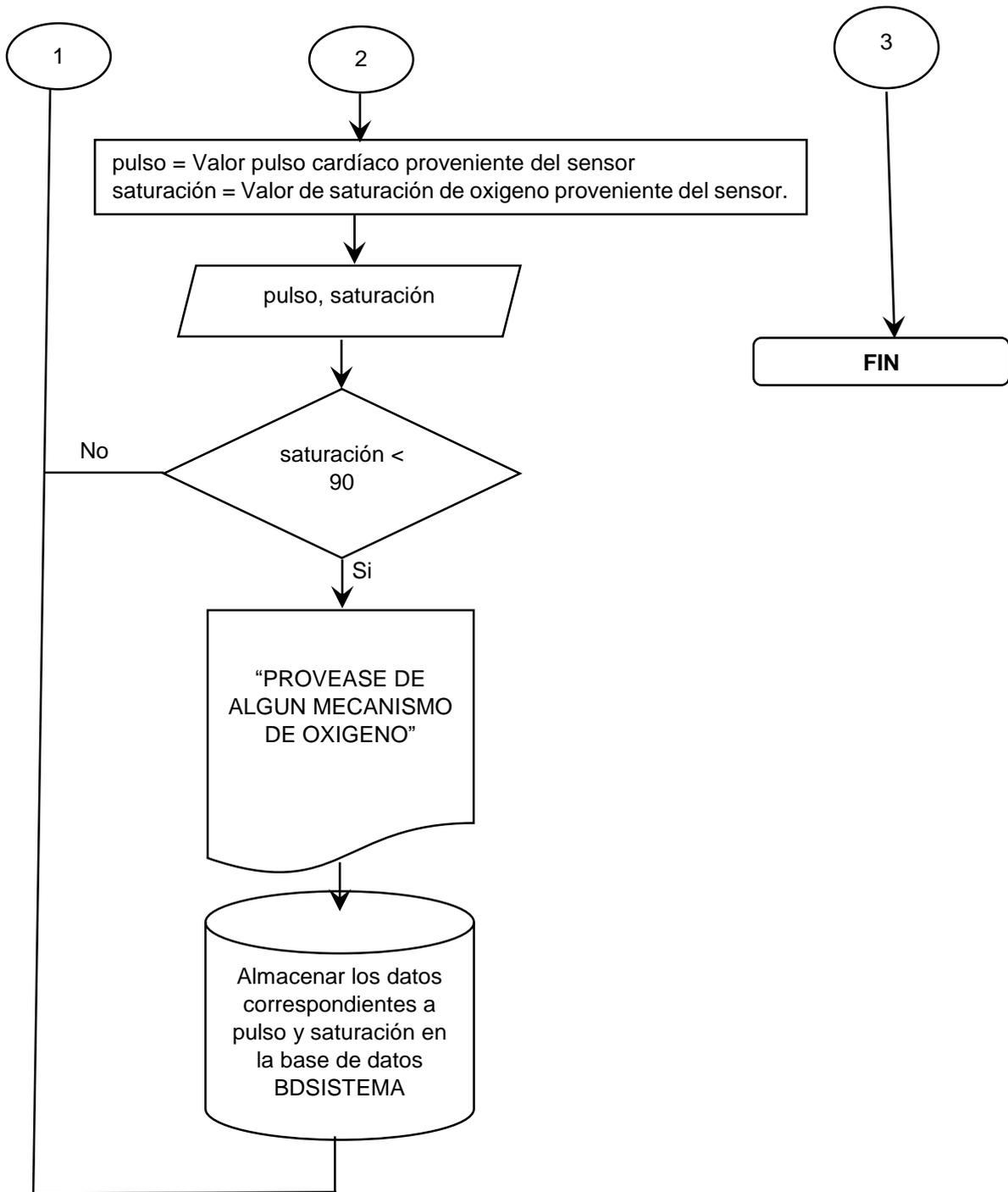


Figura 77. Diagrama de flujo del programa

4. CAPÍTULO IV: PRUEBAS DE FUNCIONAMIENTO.

Una vez concluido el diseño e implementación del sistema de monitoreo se procede a realizar las pruebas respectivas, para de esta manera poder constatar su correcto funcionamiento.

4.1. Registro y creación de Médicos.

La Figura 78 representa la tabla Doctores de la base de datos BDSISTEMA, antes de que algún registro sea ingresado.

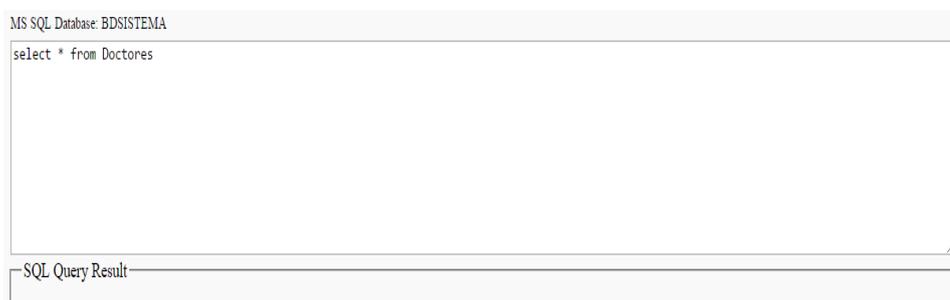


Figura 78. Tabla Doctores de la base de datos BDSISTEMA sin registros.

Como se puede observar en la Figura 78 no existe registro de algún Médico que haya sido ingresado al sistema monitoreo, por tal motivo se procederá al registro de un Médico, el cual está representado en la Figura 79.



The image shows a screenshot of a user registration form titled 'Crear Nuevo Usuario'. The form is set against a blue background with a grid pattern. At the top, there is a logo for 'UDA UNIVERSIDAD DE LAS AMÉRICAS LAUREATE INTERNATIONAL UNIVERSITY'. The form fields are as follows:

- Nombre: JAHAIRA ISABEL
- Apellidos: FONSECA
- Sexo: Mujer (dropdown menu)
- Email: oswaldo-israel@hotmail.com
- Usuario: jahaira2000
- Contraseña: [masked]

At the bottom of the form, there are two buttons: 'Iniciar sesión' and 'Crear usuario'.

Figura 79. Registro y creación de un Médico

Una vez creado el nuevo usuario, se puede comprobar que en la tabla Doctores de la base de datos BDSISTEMA, ya consta el registro con los datos del Médico que fue creado en la Figura 79, y de igual manera se envió el mensaje de bienvenida al correo personal.

MS SQL Database: BDSISTEMA

```
select * from Doctores
```

SQL Query Result

Apellidos	Nombres	Nickname	Sexo	Mail	Password
FONSECA	JAHAIRA ISABEL	jahaira2000	Mujer	oswaldo-israel@hotmail.com	19871988

Figura 80. Nuevo registro en la tabla Doctores de la base de datos BDSISTEMA.

Bienvenida al Sistema de Monitoreo y Control

 Sistema Control
Hoy, 4:34
Usted

Fotos

SISTEMA DE MONITOREO Y CONTROL 

Bienvenida Doctora **JAHAIRA ISABEL FONSECA** al sistema de monitoreo y control, recuerde que sus datos para el inicio de sesión son los siguientes:

Usuario: jahaira2000
Contraseña: 19871988

Por favor siga el siguiente link para ingresar al sistema
<http://www.serviciocontrol.somee.com/Inicio.aspx>



La salud no tiene precio y el que la arriesga es un necio

Figura 81. Mensaje personal de correo electrónico con los datos del nuevo registró.

Si se realiza una comparación entre las Figuras 79, 80 y 81 se observa que los datos ingresados en la creación y registro de Médicos, se están desplegando en la tabla de Doctores de la base de datos BDSISTEMA, y a su vez dichos datos están llegando de manera personal al correo electrónico asociado al sistema de monitoreo, por ende, se comprueba, que la creación y registro de médicos está funcionando.

4.2. Ingreso a la página web.

Una vez creado y registrado el nuevo Médico, este podrá ingresar con su usuario y contraseña, al software de comparación y gestión de datos, para posteriormente poder ingresar pacientes, comparar y modificar los datos que crea necesarios.

A) Tabla Doctores de la base de datos BDSISTEMA

```
select * from Doctores
```

SQL Query Result

Apellidos	Nombres	Nickname	Sexo	Mail	Password
FONSECA	JAHAIRA ISABEL	jahaira2000	Mujer	oswaldo-israel@hotmail.com	19871988

B) Ingreso de usuario y contraseña

C) Ingreso al sistema de monitoreo.

Figura 82. Ingreso al software de comparación y gestión de datos.

Si se realiza una comparación de los literales A), B) y C) de la Figura 82, se puede constatar, que los datos de la tabla Doctores de la base de datos BDSISTEMA, se relacionan tanto en el ingreso del usuario y contraseña como al ingreso al software de comparación y gestión de datos.

De esta manera se comprueba que el ingreso al software de comparación y gestión de datos está funcionando de manera correcta.

4.3. Registro y creación de pacientes.

La Figura 83 representa la tabla Ingreso_Paciente de la base de datos BDSISTEMA, antes de que algún registro sea ingresado.

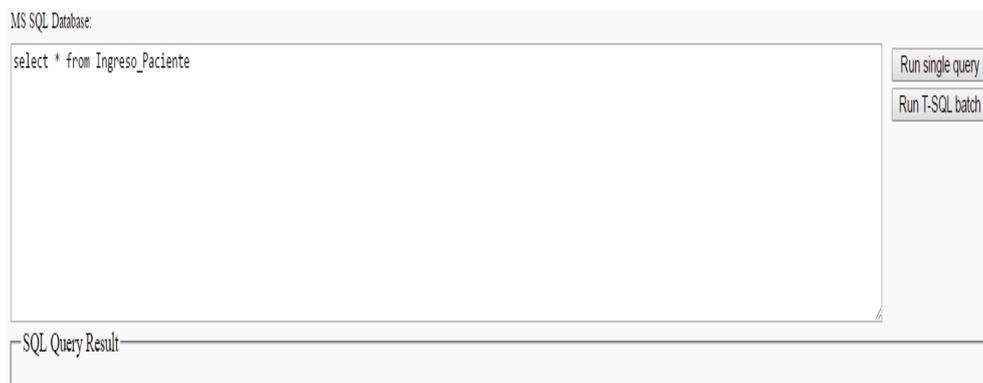
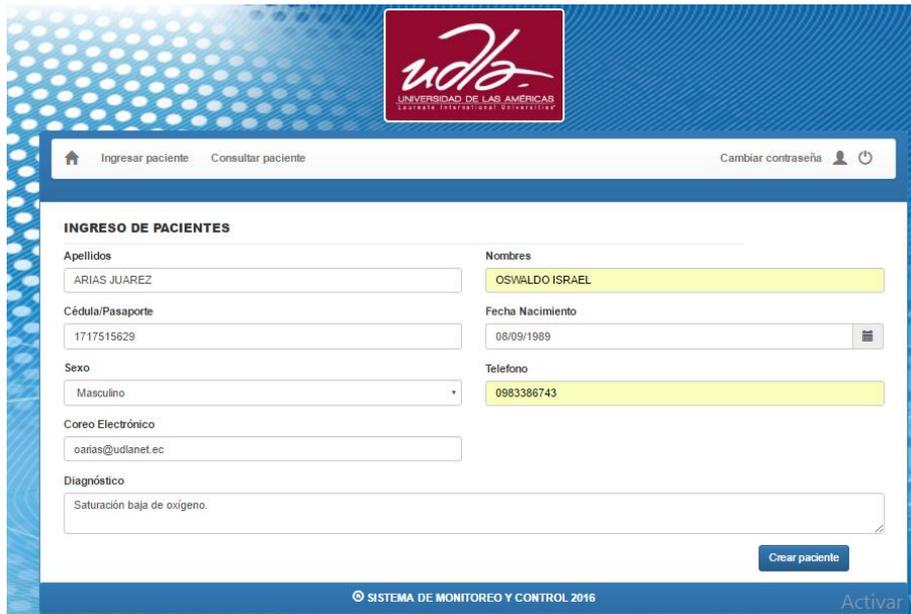


Figura 83. Tabla Ingreso_Paciente de la base de datos BDSISTEMA sin registros.

Como se puede observar en la Figura 83 no existe registro de algún paciente haya sido ingresado por algún Médico al sistema de monitoreo, por tal motivo se procederá al registro de un paciente, el cual está representado en la Figura 84.



The image shows a web interface for patient registration. At the top, there is a navigation bar with links for 'Ingresar paciente', 'Consultar paciente', and 'Cambiar contraseña'. The main form is titled 'INGRESO DE PACIENTES' and contains the following fields:

Field	Value
Apellidos	ARIAS JUAREZ
Nombres	OSWALDO ISRAEL
Cédula/Pasaporte	1717515629
Fecha Nacimiento	08/09/1989
Sexo	Masculino
Telefono	0983386743
Correo Electrónico	oarias@udlanet.ec
Diagnóstico	Saturación baja de oxígeno.

A 'Crear paciente' button is located at the bottom right of the form. The footer of the page includes the text '© SISTEMA DE MONITOREO Y CONTROL 2016' and 'Activar V'.

Figura 84. Registro y creación de un paciente.

Una vez creado el paciente, se puede comprobar que en la tabla Ingreso_Paciente de la base de datos BDSISTEMA, ya consta el registro con los datos del paciente que fue creado por el médico, y de igual manera se envió el mensaje de bienvenida al correo personal, con el aplicativo Android adjunto, para su descarga, e instalación en el dispositivo móvil.

A) Tabla Doctores de la base de datos BDSISTEMA

MS SQL Server BDSISTEMA

select * from Doctores

Run single query
Run T-SQL batch

SQL Query Result

Apellido	Nombre	Nickname	Sexo	Mail	Password
PONSECA, JAHARA ISABEL	juarez2000	oswaldo-israel	masculino	oswaldo-israel@bdsistem	1971981

B) Tabla Ingreso_Paciente de la base de datos BDSISTEMA

MS SQL Server

select * from Ingreso_Paciente

Run single query
Run T-SQL batch

SQL Query Result

Nickname_Medico	Apellido	Nombre	Cédula	Sexo	Fecha nacimiento	Mail	Teléfono	Diagnostico	Verificación
juarez2000	ARIAS JUAREZ	OSWALDO ISRAEL	171751923	Masculino	08/09/1989 (00:00:00)	oswaldo@bdsistem.ec	0983330740	Atención bajo de atención	171751923

C) Mensaje de bienvenida con archivo adjunto

Sistema Control - sistema_control@hotmail.com

para MI - X

SISTEMA DE MONITOREO Y CONTROL

Bienvenido **OSWALDO ISRAEL ARIAS JUAREZ** al sistema de monitoreo y control

Adjunto encontrará el software a descargar en su dispositivo Android

La salud no tiene precio y el que la arriesga es un necio


→ Aplicativo Android adjunto

Figura 85. Proceso de creación de nuevo paciente por médico tratante.

Si se realiza una comparación y relación de los literales A), B) y C) de la Figura 83, se puede constatar, que la creación de los pacientes y el envío de mensajes se relaciona con la tabla Doctores de la base de datos BDSISTEMA. De esta manera se comprueba que el ingreso de los pacientes y el envío de los mensajes con el archivo adjunto funcionan de manera correcta.

4.4. Recuperación de contraseña.

La recuperación de contraseña es un valor agregado que se le puso al sistema de monitoreo, con el objetivo de que, si el Médico no recuerda su contraseña, puede generar un pedido de envío de usuario y contraseña a su correo electrónico. Las Figuras 86, 87 y 88 representan lo anteriormente expuesto.



MS SQL Database: BDSISTEMA

```
select * from Doctores
```

Run single query
Run T-SQL batch

SQL Query Result

Apellidos	Nombres	Nickname	Sexo	Mail	Password
FONSECA	JAHAIRA ISABEL	jahaira2000	Mujer	oswaldo-israel@hotmail.com	19871988

Figura 86. Tabla Doctores de la base de datos BDSISTEMA.



UDLA
UNIVERSIDAD DE LAS AMÉRICAS
CARRERA INTERDISCIPLINARIA DE INGENIERÍA

Recuperar Datos de Usuario

Usuario

Email

Figura 87. Solicitud de envío de contraseña al correo electrónico.

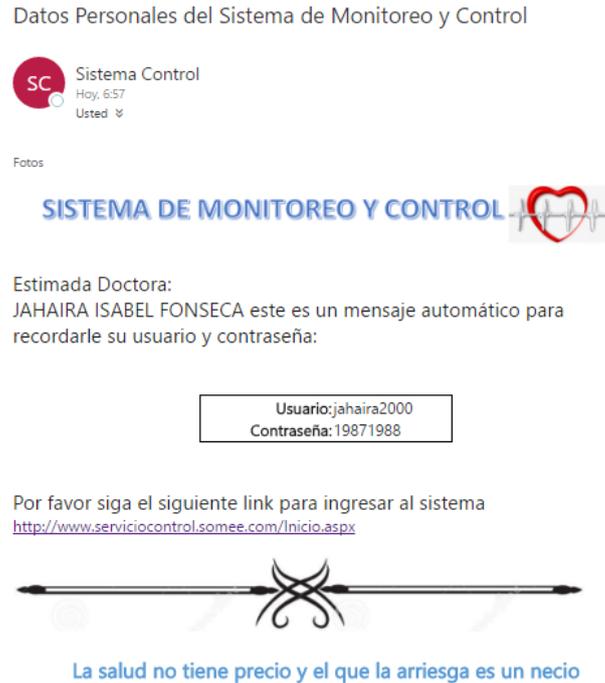


Figura 88. Mensaje recibido el usuario y contraseña del médico tratante

Si se relacionan la Figura 86, 87 y 88, se puede constatar que la información recibida en el mensaje de correo electrónico representado en la Figura 86 coincide con la información almacenada en la tabla Doctores de la base de datos BDSISTEMA. De esta manera queda comprobado que la opción recuperar contraseña funciona de manera correcta.

4.5. Cambio de contraseña.

Al igual que recuperación de contraseña, el cambio de contraseña es un valor agregado que se le puso al sistema de monitoreo, con el objetivo de que, si el Médico no está a gusto con su contraseña de ingresa al software de comparación y gestión de datos, puede proceder a generar un cambio de contraseña. La Figura 89 representa lo anteriormente expuesto.

A) Tabla Doctores de la base de datos BDSISTEMA con la antigua contraseña.

Ms SQL Enterprise

select * from Doctores

Run single query
Run T-SQL batch

SQL Query Result

Apellidos	Nombres	Nickname	Sexo	Mail	Password
FONSECA	JAHAIRA ISABEL	jahaira2000	Mujer	oswaldo-israel@hotmail.com	19871988

B) Solicitud de cambio de contraseña

Logo: **uda** UNIVERSIDAD DE LAS AMÉRICAS

Home | Ingresar paciente | Consultar paciente | Cambiar contraseña

Cambio de Contraseña

Por favor Doctora JAHAIRA ISABEL FONSECA ingrese su nueva contraseña en los espacios en blanco

Contraseña: 12345678

Repetir Contraseña: 12345678

Cambiar contraseña | Cancelar

© SISTEMA DE MONITOREO Y CONTROL 2016

C) Tabla Doctores de la base de datos BDSISTEMA con la nueva contraseña.

SQL Query Result

Apellidos	Nombres	Nickname	Sexo	Mail	Password
FONSECA	JAHAIRA ISABEL	jahaira2000	Mujer	oswaldo-israel@hotmail.com	12345678

Figura 89. Proceso de cambio de contraseña.



Figura 90. Mensaje personal de correo electrónico con los nuevos datos de usuario y contraseña.

Si se observa los literales A) y C) de la Figura 89 se puede constatar que la contraseña cambio en la tabla Doctores de la base de datos BDSISTEMA, dicha contraseña será usada de ahora en adelante para el ingreso del médico al software de comparación y gestión de datos.

4.6. Ingreso al aplicativo Android.

El aplicativo Android es el software encargada de monitorear la saturación de oxígeno y pulso cardíaco del paciente, es importante tener en cuenta, que el paciente que vaya a usar este aplicativo, debe anteriormente haber sido registrado por su médico tratante y constar en la tabla de Ingreso_Paciente de la base de datos BDSISTEMA.

A) Tabla Ingreso_Paciente de la base de datos BDSISTEMA.

MS SQL Server

```
select * from Ingreso_Paciente
```

Run single query
Run T-SQL batch

SQL Query Result

Nickname Médico	Apellidos	Nombres	Cédula	Sexo	Fecha nacimiento	Mail	Teléfono	Diagnóstico	Verificación
jhuara2000	ARIAS JUAREZ	OSWALDO ISRAEL	1717515629	Masculino	08-09-1989 00:00:00	oarias@odlanet.ec	0983386743	Saturación baja de oxígeno.	1717515629

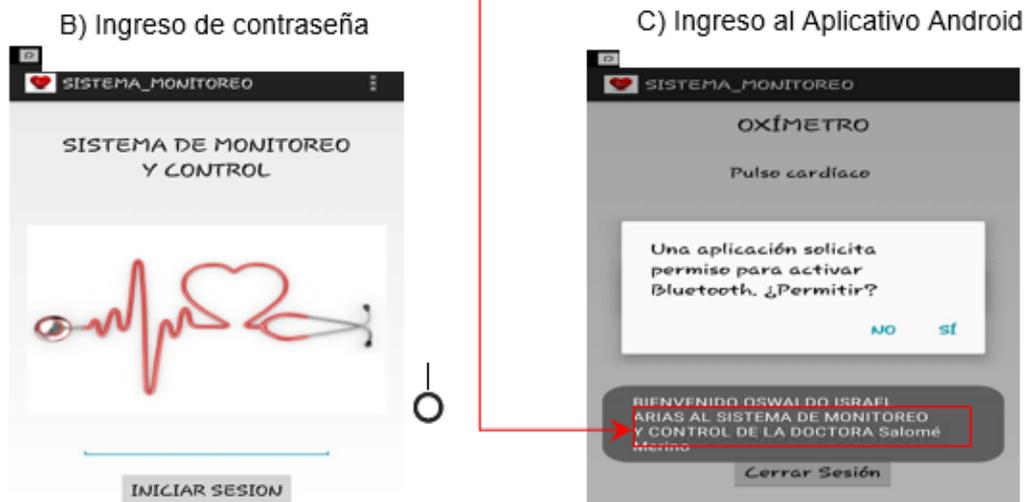


Figura 91. Proceso de ingreso al aplicativo Android

Como se puede observar en los literales A) y B) de la Figura 91 el paciente debe anteriormente haber sido registrado por su médico tratante, para que el sistema de monitoreo le permita el ingreso al aplicativo Android.

De esta manera se comprueba el funcionamiento del ingreso al aplicativo Android.

4.7. Adquisición y comparación de los datos obtenidos por el sistema de monitoreo.

Una vez que el paciente ha ingresado al aplicativo Android, se procede a la sincronización del prototipo con el aplicativo Android para el envío de los datos correspondientes a la saturación de oxígeno y pulso cardíaco.



Figura 92. Valores del pulso cardíaco y saturación de oxígeno en la sangre

Para comprobar la veracidad de los datos del sistema de monitoreo, se hará una comparación de los datos obtenidos, por un oxímetro comercial de la marca fingertip modelo JERRY-F, con 5 muestras obtenidas por el sistema de monitoreo y control, en lo que corresponde pulso cardíaco y saturación de oxígeno en la sangre en 5 personas. La tabla 21, 22, 23, 24 y 25 representan lo anteriormente expuesto.

Tabla 21.

Pruebas de funcionamiento en primera persona

	Oxímetro JERRY-F		Sistema de monitoreo y control		% Error	
	PR	SPO2	PR	SPO2	PR	SPO2
Primera muestra	72	91	73	91	1,39	0
Segunda muestra	73	89	74	88	1,36	1,12
Tercera muestra	74	92	74	90	0	2,17
Cuarta muestra	73	93	74	93	1,36	0
Quinta muestra	76	89	76	88	0	1,12

Tabla 22.

Pruebas de funcionamiento en segunda persona

	Oxímetro JERRY-F		Sistema de monitoreo y control		% Error	
	PR	SPO2	PR	SPO2	PR	SPO2
Primera muestra	75	90	74	88	1,33	2,22
Segunda muestra	77	89	75	90	2,60	1,12
Tercera muestra	70	88	70	88	0	0
Cuarta muestra	73	91	73	91	0	0
Quinta muestra	67	93	66	92	1,50	1,08

Tabla 23.

Pruebas de funcionamiento en tercera persona

	Oxímetro JERRY-F		Sistema de monitoreo y control		% Error	
	PR	SPO2	PR	SPO2	PR	SPO2
Primera muestra	67	88	66	88	1,50	0
Segunda muestra	65	91	65	90	0	1,09
Tercera muestra	69	89	68	88	1,45	1,12
Cuarta muestra	70	90	71	90	1,43	0
Quinta muestra	69	87	69	85	0	2,30

Tabla 24.

Pruebas de funcionamiento en cuarta persona

	Oxímetro JERRY-F		Sistema de monitoreo y control		% Error	
	PR	SPO2	PR	SPO2	PR	SPO2
Primera muestra	72	91	71	90	1,39	1,09
Segunda muestra	70	88	70	90	0	2,27
Tercera muestra	82	85	80	86	2,44	1,18
Cuarta muestra	70	88	69	89	1,43	1,14
Quinta muestra	67	91	67	91	0	0

Tabla 25.

Pruebas de funcionamiento en quinta persona

	Oxímetro JERRY-F		Sistema de monitoreo y control		% Error	
	PR	SPO2	PR	SPO2	PR	SPO2
Primera muestra	80	95	80	94	0	1,05
Segunda muestra	85	94	84	93	1,18	1,06
Tercera muestra	82	92	80	92	2,44	0
Cuarta muestra	79	95	81	95	2,53	0
Quinta muestra	81	91	81	90	0	0

4.8. Registro de eventos para el control continuo del paciente.

Los valores del pulso cardíaco y de la saturación de oxígeno, se despliegan en el dispositivo móvil, pero si existe alguna anomalía en la saturación de oxígeno en la sangre se desplegará una alarma a sonora, y los valores responsables de la activación se almacenarán en la tabla Control_Paciente de la tabla BDSISTEMA.

La Figura 93 representa la tabla Control_Paciente de la tabla BDSISTEMA antes de que algún registro sea almacenado.



Figura 93. Tabla Control_Paciente de la base de datos BDSISTEMA sin registros.

Las Figuras 94 y 95 representan el proceso del almacenamiento de los eventos en la tabla Control_Paciente de la base de datos BDSISTEMA cuando la alarma del aplicativo móvil ha sido activada.



Figura 94. Activación de la alarma en el aplicativo Android.

MS SQL Database

Select * from Control_Paciente

Run single query
Run T-SQL batch

SQL Query Result

Nickname M	Cédula P	Paciente	Fecha Hora	Saturación	Pulso	Observación	idPaciente
jahaira2000	1717515629	ARIAS JUAREZ OSWALDO ISRAEL	28/11/2016 3:32:38	72.0 SPO2	83.0 PR	POSEE LA SATURACION BAJA	143

Figura 95. Nuevo registró en la tabla Control_Paciente de la base de datos BDSISTEMA.

Como se puede observar en la Figura 95, ya existe un registro en la tabla Control_Paciente de la base de datos BDSISTEMA, que fue creado después de que se activó la alarma en el dispositivo móvil representado en la Figura 94, si se revisa la Figura 93, se puede constatar que antes de que la alarma fuera activada el dispositivo móvil, no existía ningún registro en la tabla Control_Paciente de la base de datos BDSISTEMA.

Con el proceso anteriormente detallado se comprueba que el almacenamiento de los eventos cuando la alarma del dispositivo móvil es activada, están siendo almacenados en la tabla Control_Paciente de la base de dato BDSISTEMA.

5. CAPÍTULO V: ANALISIS COSTO-BENEFICIO.

Este capítulo está destinado a la comparación del sistema de monitoreo con otros dispositivos existentes en el mercado que realizan la misma función con el objetivo de poder establecer diferencias, ventajas y costos.

Para cumplir lo anteriormente mencionado se han escogido 2 oxímetros comerciales, para de esta manera evaluar sus funciones y costos, frente al sistema de monitoreo.

- **Iincare Bluetooth Oximeter.**

En la tabla 26 se detallan las principales características del oxímetro de pulso Iincare Bluetooth Oximeter.

Tabla 26.

Características del oxímetro Iincare Bluetooth Oximeter

CARACTERISTICAS – ILINCARE BLUETOOTH OXIMETER	
	Integrado con SPO2 sonda y el módulo de visualización de procesamiento
	SPO2, PR y PI display.
	Indicación del valor de la frecuencia Del Pulso, display gráfico de barras
	Comunicación bluetooth 4.0 con software inteligente para dispositivo movil
	Integrado con SPO2 sonda y el módulo de visualización de procesamiento.
Costo	\$ 92,37

Tomado de (Aliexpress s.f.)

- **Oxímetro wirelles bluetooth android apple gima.**

En la tabla 27 se detallan las principales características del oxímetro de pulso wirelles bluetooth android apple gima.

Tabla 27.

Características del oxímetro wirelles bluetooth apple gima.

CARACTERISTICAS – WIRELLES BLUETOOTH APPLE GIMA	
Medir su saturación de oxígeno arterial y ritmo cardíaco desde un teléfono móvil	
Registra la tasa de SpO2 y pulso	
Indicación del valor de la frecuencia Del Pulso, display gráfico de barras	
Comunicación bluetooth 4.0 con software inteligente para dispositivo móvil	
Provisto de una cuerda de seguridad y cable de carga USB	
Costo	\$ 92,28

Tomado de (Casaesport s.f.)

- **Sistema de monitoreo y control.**

En la tabla 28 se detallan los accesorios y agregados con sus respectivos costos que fueron usados para el diseño e implementación del sistema de monitoreo y control.

Tabla 28.

Tabla de costos del sistema de monitoreo y control

DETALLE	COSTO
Sensor nellcor ds-100A 9 pines	\$ 70
Microcontrolador Atmega 164 P	\$ 7
Circuito integrado regulador l7805cv - 5v	\$ 0.50
Bluetooth HC-06	\$ 9.00
Baterías 3,5 V para alimentación	\$ 22.00
Alojamiento web	\$19.95
Diseño de página WEB	\$150.00
Diseño de aplicativo Android	\$150.00
Costo	\$428.45

En la tabla 29 se detallan las principales características del sistema de monitoreo y control.

Tabla 29.

Características del sistema de monitoreo y control.

CARACTERÍSTICAS – SISTEMA MONITOREO Y CONTROL	
Medir su saturación de oxígeno arterial y ritmo cardíaco	
Indicación visual del pulso cardíaco y saturación de oxígeno en un dispositivo móvil.	
Aviso con alarma sonora cuando existe alguna anomalía en la saturación de oxígeno.	
Comunicación bluetooth 4.0 con software inteligente para dispositivo móvil	
Almacenamiento de registros correspondientes a pulso cardíaco y saturación de oxígeno en la sangre en un alojamiento web.	
Página web con interfaz amigable para que el médico tratante pueda tener acceso a los registros de sus pacientes.	
Costo	\$ 446.95

La tabla 30 detalla un resumen comparativo de los 3 dispositivos detallados anteriormente.

Tabla 30.

Tabla comparativa de los 3 dispositivos de medición.

	ILINCARE BLUETOOTH OXIMETER	WIRELLES BLUETOOTH ANDROID APPLE GIMA	SISTEMA DE MONITOREO Y CONTROL
Medición del pulso cardíaco	√	√	√
Medición de la saturación de Oxígeno	√	√	√
Aplicativo para dispositivo móvil	√	√	√
Indicador visual del pulso cardíaco y saturación de oxígeno en la sangre	√	√	√
Alarma sonora de notificación	—	—	√
Diseño de página web para el gestiona miento de los datos	—	—	√
Almacenamiento de los registros y datos en un alojamiento web	—	—	√
Costo	\$ 92,37	\$ 92,28	\$ 446,95

Como se puede observar en la tabla 30 el costo del sistema de monitoreo y control es más elevado, pero posee más ventajas con respecto a los otros dispositivos, ya que, al tener una interfaz gráfica y un alojamiento web, los valores del pulso cardíaco y saturación de oxígeno en la sangre se almacenan, y no se pierden, para que de esta manera el Médico de cabecera pueda tener acceso a ellos y tener un control más pertinente de sus pacientes a través de la página web.

6. CONCLUSIONES Y RECOMENDACIONES

Tras desarrollo e implementación del sistema de monitoreo y control se debe presentar las conclusiones que se obtuvo y las recomendaciones que se pueden dar.

En este apartado se presentará la información anteriormente detallada, para que cualquier persona pueda entender de qué se trató el proyecto y lo que se consiguió tras su realización.

6.1. Conclusiones.

El estudio y análisis previo de los requerimientos para la implementación y desarrollo del sistema de monitoreo para la medición de pulso cardíaco y saturación de oxígeno en la sangre, fue de vital importancia ya que de esa manera se pudo obtener los materiales, herramientas, equipos y fuentes de información los cuales fueron sometidos, a un análisis, comparación y selección para rescatar aquellos recursos que sirvieron de base y ayuda en el diseño del sistema de monitoreo.

Todo diseño y construcción de un oxímetro de pulso debe tener también un software de gestión y control que permita almacenar datos y registros del pulso cardíaco y de la saturación oxígeno en la sangre de una persona, para que de esta manera un médico tratante puede ingresar, observar, analizar y dar su punto de vista. Este aspecto es muy importante para poder llevar un correcto control Médico, los oxímetros comerciales no cuenta con un software de gestión de datos, es decir solo permiten apreciar visualmente los valores correspondientes al pulso cardíaco y saturación de oxígeno en la sangre, y si existe alguna anomalía en estos valores, no hay manera de saberlo y los datos se pierden sin saber, ¿cuándo? y el ¿por qué? se dió esta anomalía en los valores.

Con el sistema de monitoreo se solventa este inconveniente, ya que, al tener un software de gestión de datos, los valores de pulso cardíaco y saturación de oxígeno que no sean normales en un paciente se almacenarán en una base de datos, para que un médico tratante puede tener acceso a ellos, y llevar un mejor control de sus pacientes.

El sistema de monitoreo para la medición del pulso cardíaco y saturación de oxígeno en la sangre está constituido por un dispositivo electrónico, un aplicativo Android y un software para el gestiona miento y control de datos, los cuales una vez culminados fueron sometidos a pruebas de funcionamiento para saber si el proyecto cumple con el objetivo planteado; la tabla 21 muestra una comparación de los datos obtenidos por un Oxímetro comercial modelo JERRY-F y el sistema de monitoreo y control, obteniendo un margen de error en el pulso cardíaco entre (0 – 2.60) % y en la saturación de oxígeno en la sangre entre (0 – 2.30)%, existen pequeñas variaciones, pero eso se debe a como está colocado el dedo índice en el sensor, de esta manera se concluye que el sistema de monitoreo y control se ajusta y cumple con los objetivos planteados.

La ventaja de crear un software de gestiona miento y control de datos en C#asp.net como lenguaje de programación es que existen varios alojamientos web gratuitos que trabajan o son compatible con asp.net, para que de esta manera el software de gestiona miento y control de datos puede ser accesible mediante cualquier dispositivo conectado a internet, en cuanto a costos de hardware, estos fueron más elevados que software, ya que para el diseño del prototipo, se requirieron materiales y herramientas físicas que fueron adquiridos en el mercado; en la tabla 27 se observa que en la comparativa de costos, el sistema de monitoreo tiene un costo más elevado comparado con los 2 dispositivos comerciales, pero posee más características que son fundamentales e importantes, en un sistema de monitoreo y control.

6.2. Recomendaciones.

Actualmente el ser humano ha descuidado por ciertos motivos el cuidado de su salud, es por eso que se recomienda el uso del sistema de monitoreo y control, ya que uno de los aspectos importantes en la vida humana es el cuidado del pulso cardíaco y saturación de oxígeno en la sangre ya que, si no se tiene un control previo de estos dos aspectos, se puede llegar a tener graves problemas de salud, por tal motivo se recomienda el uso del sistema de monitoreo y control del pulso cardíaco y saturación de oxígeno en la sangre.

Es recomendable que antes de usar el software de gestión y control de datos se realice una inspección previa de cómo está diseñado el software y como se debe ingresar correctamente los datos para su previo análisis.

La interfaz gráfica de la página web debe ser amigable y entendible para el usuario final, este aspecto es muy importante y muy recomendable, ya que, si una página web cumple con todos los requisitos pedidos o deseados, pero no es entendible, por el usuario, va a causar severos problemas y por ende la página web desde el punto de vista del usuario final va estar mal diseñada.

REFERENCIAS

- Aliexpress. (s.f.). *Bluetooth Oximeter*. Recuperado el 8 de julio del 2016 de:
<https://es.aliexpress.com/item/Free-Shipping-CE-FDA-Bluetooth-Fingerip-pulse-oximeter-SPO2-meter-heart-rate-monitor-for-android-IOS/32371212114.html?spm=2114.43010208.4.15.JvZZ38>
- Alvarado, S. y Juárez D. (2012). *Redes de área corporal en el cuidado de la salud*. Recuperado el 19 de octubre del 2015 de:
<http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/2568/Tesis.pdf?sequence=1>
- Android OS. (s.f.). *Android características*. Recuperado el 09 de febrero del 2016 de: <http://androidos.readthedocs.org/en/latest/data/caracteristicas/>
- Atmel. (s.f.). *Atmel*. Recuperado el 11 de febrero del 2016 de:
<http://www.mouser.com/ds/2/36/Atmel-8011-8-bit-AVR-microcontroller-ATmega164P-32-356485.pdf>
- Betancur, L. (2011). *Redes de área corporal. Una perspectiva al futuro desde la investigación*. Recuperado el 21 de septiembre del 2015 de:
http://www.icesi.edu.co/revistas/index.php/sistemas_teleomatica/articulo/viewFile/1027/1052
- Blogsobrebluetooth. (s.f.). *¿Qué es Bluetooth?*. Recuperado el 21 de septiembre del 2015 de: <http://blogsobrebluetooth.blogia.com/>
- BotScience (2014). *Módulo Bluetooth slave HC-06*. Recuperado el 03 de febrero del 2016 de:
http://botscience.net/store/index.php?route=product/product&product_id=69
- Cables y sensores. (s.f.). *Cables y sensores*. Recuperado el 31 de enero del 2016 de: <http://www.cablesysensores.com/>
- Carrillo, M. (2006). *Bluetooth*. Recuperado el 21 de septiembre del 2015 de:
<http://www.iit.upcomillas.es/pfc/resumenes/4505a8eba9cfc.pdf>

- Casesport. (s.f.). *OXÍMETRO WIRELLES BLUETOOTH ANDROID APPLE GIMA*. Recuperado el 28 de noviembre del 2016 de: <http://www.casaesport.it/catalogos/PULSO-OXIMETRO-PULSO-OXIMETRO-OX-METRO-WIRELLES-BLUETOOTH-ANDROID-APPLE-GIMA>
- CCM. (s.f.). *Modos de funcionamiento Wifi (802.11 o Wi-Fi)*. Recuperado el 21 de septiembre del 2015 de: <http://es.ccm.net/contents/791-modos-de-funcionamiento-wifi-802-11-o-wi-fi>
- Cemeblog. (s.f.). *Tipo de estándares Wi-fi*. Recuperado el 21 de septiembre del 2015 de: <http://blog.cemebe.info/tag/wifi/>
- Cenetec. (s.f.). *Pulso Oximetría*. Recuperado el 08 de febrero del 2016 de: http://its.uvm.edu/medtech/module_es.html
- Cibermedico. (s.f.). *BODY AREA NETWORKS: Un gran futuro en la monitorización médica y en la salud de consumo*. Recuperado el 21 de septiembre del 2015 de: <http://cibermedico.chinatantan.com/?m=201111>
- Covidien. (2012). *Sistema respiratorio adjunto de monitorización de pacientes*. Recuperado el 21 de septiembre del 2015 de: <http://www.covidien.com/imageServer.aspx/doc261834.pdf?contentID=35934&contenttype=application/pdf>
- Fundación Española del corazón. (s.f.). *Frecuencia cardíaca*. Recuperado el 17 de enero del 2016 de: http://www.fundaciondelcorazon.com/prevencion/riesgo_cardiovascular/frecuencia-cardiaca.html
- Madrid. (s.f.). *Estos son los efectos de la falta de oxígeno en nuestro cuerpo*. Recuperado el 31 de enero del 2016 de: http://www.teinteresa.es/salud/efectos_falta_oxigeno_0_1135088191.html
- Manosalvas, S (s.f.). *Lenguajes de Programación cuadro comparativo*. Recuperado el 09 de febrero del 2016 de: <http://blog.buhoos.com/lenguajes-de-programacion-cuadro-comparativo/>

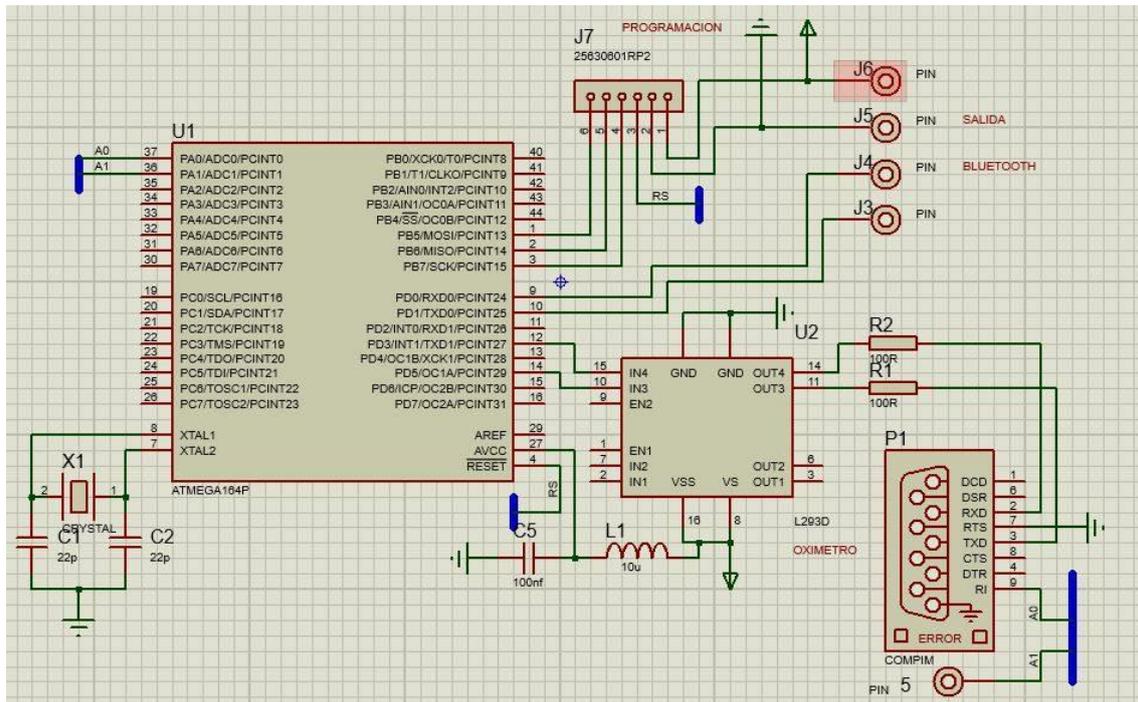
- Martínez, H. (2014). *Recomendaciones de seguridad para tu wifi*. Recuperado el 21 de septiembre del 2015 de: <http://solucionesinformatica-sepi.blogspot.com/2016/04/principales-factores-de-riesgo-y-recomendaciones-de-seguridad-para-tu-WiFi-SEPI-Consultores.html>
- MetroMéxico. (s.f.). *Estándares Inalámbricos*. Recuperado el 21 de septiembre del 2015 de: <http://www.metrologicmexico.com/elemento.php?var=258>
- Microcontroladores. (s.f.). *Convertidor Analógico Digital AVR*. Recuperado el 23 de mayo del 2016 de: <http://microcontroladores-mrelberni.com/convertidor-analogico-digital-avr/>
- Microsoft. (2010). *Implementación de infrarrojos*. Recuperado el 21 de septiembre del 2015 de: [https://msdn.microsoft.com/es-ec/library/cc756996\(v=ws.10\).aspx](https://msdn.microsoft.com/es-ec/library/cc756996(v=ws.10).aspx)
- Microsoft. (2015). *Visual C#*. Recuperado el 09 de febrero del 2016 de: <https://msdn.microsoft.com/es-ec/library/kx37x362.aspx>
- Millan, R. (2012). *La explosión de Bluetooth: tecnología, aplicaciones y productos*. Recuperado el 21 de septiembre del 2015 de: <http://www.ramonmillan.com/tutoriales/bluetooth.php>
- Mouser Electronics. (s.f.). *Mouser Electronics*. Recuperado el 18 de mayo del 2016 de: <http://www.mouser.ec/ProductDetail/STMicroelectronics/L7805CV/?qs=9NrABI3fj%2FqplZAHiYUxWg%3D%3D>
- Mundo informático. (s.f.). *Sql Server conceptos básicos*. Recuperado el 10 de febrero del 2016 de: <http://rjprogramacion.blogspot.com/2013/06/sql-server-conceptos-basicos.html>
- Nellcor. (s.f.). *Oximax DS-100A*. Recuperado el 31 de enero del 2016 de: http://www.frankshospitalworkshop.com/equipment/documents/pulse_oximeter/user_manuals/Nellcor_Durasensor.pdf
- Nieto, A. (2011). *¿Qué es Android?* Recuperado el 09 de febrero del 2016 de: <http://www.xatakandroid.com/sistema-operativo/que-es-android>

- Noguerol, C. y Gonzáles A. (2012). *Pulsioximetría*. Recuperado el 23 de agosto del 2015 de: <https://www.fisterra.com/material/tecnicas/pulsioximetria/pulsioximetria.pdf>
- Pecheur. (s.f.). Cinturón gramin frecuencímetro cardio. Recuperado el 09 de febrero del 2016 de: <https://msdn.microsoft.com/es-es/library/kx37x362.aspx>
- Prieto, J. (2011). *Introducción a los sistemas de comunicación inalámbrica*. Recuperado el 21 de septiembre del 2015 de: [https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_\(Modulo_1\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_1).pdf)
- Prometec. (s.f.). *Prometec*. Recuperado el 11 de febrero del 2016 de: <http://www.prometec.net/hbridge/>
- Ronquillo, J. y Salgado, P. (2013). *Diseños y construcción de un oxímetro de pulso portátil*. Recuperado el 10 de febrero del 2016 de: <http://dspace.uazuay.edu.ec/bitstream/datos/3291/1/10065.pdf>
- Samsung. (s.f.). *Samsung*. Recuperado el 18 de mayo del 2016 de: http://www.samsung.com/sg/consumer/mobile_devices/accessories/battery/EB-B500AEBECWW
- Scielo. (s.f.). *Ingeniería y Desarrollo*. Recuperado el 2 de julio del 2015 de: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0122-34612009000100011
- Scribd. (s.f.). *Teorema de Muestreo de Nyquist*. Recuperado el 31 de enero del 2016 de: <https://es.scribd.com/doc/67341915/Teorema-de-Muestreo-de-Nyquist>
- Sensormed. (s.f.). *Pulso Oxímetro*. Recuperado el 03 de febrero del 2016 de: <http://www.jomedical.com/accesorios?format=raw&task=download&fid=189>

- SlideShare. (2010). *Redes Inalámbricas*. Recuperado el 21 de septiembre del 2015 de: <http://es.slideshare.net/SuperFonso/redes-inalambricas-5108220>
- Suatunce, J. (2014). *Diseño e implementación de un dispositivo de monitorización de oximetría*. Recuperado el 03 de febrero del 2016 de: <http://repositorio.espe.edu.ec/bitstream/21000/8876/1/T-ESPE-048032.pdf>
- Suatunce, J. (2014). *Diseño e implementación de un dispositivo de monitorización remota*. Recuperado el 21 de septiembre del 2015 de: <http://repositorio.espe.edu.ec/bitstream/21000/9350/1/T-ESPEL-ENI-0339.pdf>
- Tecmikro. (s.f.). *L293B – L293D: driver para motores*. Recuperado el 12 de febrero del 2016 de: <http://www.programarpicenc.com/articulos/l293d-l293b-driver-para-motores-cc-dc/>
- Tecno revolución. (s.f.). *Redes de área corporal (Body Area Network)*. Recuperado el 21 de septiembre del 2015 de: <http://tecnorevolucion.blogspot.com/2012/12/redes-de-area-corporal-body-area-network.html>
- Urespuestas. (2015). *¿Cuánto latidos por minuto son normales?* Recuperado el 22 de mayo del 2016 de: <http://microrespuestas.com/cuantos-latidos-por-minuto-son-los-normales/>
- Vallejo, M. (2011). *Signos vitales*. Recuperado del 23 de agosto del 2015 de: <http://es.slideshare.net/JuanDiego16/signos-vitales-9321597>
- Ventura, V (2015). *Principio de funcionamiento del oxímetro para monitorización del pulso*. Recuperado el 18 de mayo del 2016 de: <http://polaridad.es/monitorizacion-sensor-pulso-oximetro-frecuencia-cardiaca/>
- Vida embebida. (2014). *AVR programar*. Recuperado el 5 de mayo del 2016 de: <https://vidaembebida.wordpress.com/2014/08/17/avr-programacion-en-c-09-timercounter0-del-atmega16-parte1/>

ANEXOS

ANEXO 1: Diagrama de circuito del prototipo



ANEXO 2: Programación del microcontrolador Atmega 164 P

```
#include <stdlib.h> // for itoa() call
#include <stdio.h>
#include <avr/io.h>
#include <avr/pgmspace.h>
#include <inttypes.h>
#include <avr/interrupt.h>
#include <avr/eeprom.h>
#ifndef F_CPU
#define F_CPU 2000000UL /* quartz with 3,6864 MHz */
#endif
#include <util/delay.h>
#include "Usart.h"
ISR(ADC_vect);
#define EnviarMatriz      0 // Si es igual a 1 Envia los datos totales leidos
#define Adc_resolucion_Op 1 //SIEMPRE IGUAL A UNO
```

```

#define PD3_LOW   PORTD &=~(1<<3)
#define PD5_LOW   PORTD &=~(1<<5)
#define PD5_HIGH  PORTD |= (1<<5)
#define PD3_HIGH  PORTD |= (1<<3)
#define MuestrasTotales  56
#define DatosxSeg  14
//14 datos por segundo
//TRAMAS TOTALES
int Led_R_T[MuestrasTotales];
int Led_I_T[MuestrasTotales];
//TRAMAS DIVIDIDAS EN GRUPO DE 14 MUESTRAS POR SEGUNDO
int Led_R1s[DatosxSeg];
int Led_R2s[DatosxSeg];
int Led_R3s[DatosxSeg];
int Led_R4s[DatosxSeg];
int Led_I1s[DatosxSeg];
int Led_I2s[DatosxSeg];
int Led_I3s[DatosxSeg];
int Led_I4s[DatosxSeg];
#if (Adc_resolucion_Op)
    uint16_t adc_valor = 0;
    void adc_init_op(void);
    void adc_leer_canal_op(void);
    #define adc_gan10  1

    //DETERMINA LA VELOCIDAD DE MUESTREO
    #define adc_fec4  0
    #define adc_fec8  1
    #define adc_fec16  0
#endif

```

```
//CALCULA EL PROMEDIO DE ROJO E INFRARROJO
```

```
uint32_t Val_promedio=0;
uint16_t Final_promedio=0;
//-----
uint8_t Tiempo_led=0;
uint8_t Opciones_led=0; //permite que los leds se enciendan alternadamente
const uint16_t TiempoMuestreo=70;//separacion en el tiempo entre cada
muestra
char buffer [6];//PERMITE ENVIAR LA TRAMA CORRESPONDIENTE A UN
NUMERO
void Matriz_ADC(uint8_t PosMatriz);
void AnalisisNumerico(void);
void CalMayorMenor(int Mtr_Mn[]);
int main(void)
{
    //-----
    USART_Init(BAUD_PRESCALE);//INICIA COMUNICACION SERIAL
    char String_1[]="INTEGRADO CONFIG MEGA\0";// String a ser enviada
    UTransmit_String(String_1);
    //-----

    //DEFINIENDO SALIDAS
    DDRD|= 1<<7;
    DDRD|= 1<<5;
    DDRD|= 1<<3;

    //Valor inicial salidas de CERO logico
    PD3_LOW;
    PD5_LOW;
    //Valor inicial salidas de UNO logico
    PORTD |=(1<<7);
```

if (Tiempo_led>50){Tiempo_led=4;}//en caso que se inicie la grabacion
 estos valores aseguran el funcionamiento

if (Tiempo_led==0){Tiempo_led=4;}//en caso que se inicie la grabacion
 estos valores aseguran el funcionamiento

```
#if (Adc_resolucion_Op)
    adc_init_op();
#endif
```

//----- VALORES DE CONFIGURACION NO PUEDEN SER VARIADOS

```
Opciones_led = 2;
```

```
Tiempo_led = 4;
```

```
//-----
```

```
while (1)
```

```
{
```

```
//LEE LAS MUESTRAS DURANTE 4 SEGUNDOS DEL LED
```

ROJO

```
Opciones_led=0;PD3_LOW;PD5_HIGH;//PARA LED ROJO
```

```
for( uint8_t Muestreo = 0; Muestreo < MuestrasTotales+1;
```

```
Muestreo++) {Matriz_ADC(Muestreo);}
```

```
//LEE LAS MUESTRAS DURANTE 4 SEGUNDOS DEL LED
```

INFRARROJO

```
Opciones_led=1;PD3_HIGH;PD5_LOW;//PARA INFRARROJO
```

```
for( uint8_t Muestreo = 0; Muestreo < MuestrasTotales+1;
```

```
Muestreo++) {Matriz_ADC(Muestreo);}
```

```
//DETERMINA LOS MAXIMOS - MINIMOS Y EL PROMEDIO
```

TOTAL

```
AnalisisNumerico();
```

```
    }  
    return 0;  
}  
  
void Matriz_ADC (uint8_t PosMatriz)  
{  
    #if (Adc_resolucion_Op)  
        adc_leer_canal_op();  
    #endif  
  
    _delay_ms(TiempoMuestreo);  
  
    if (PosMatriz > 0)  
    {  
        if(Opciones_led==0)  
        {  
            Led_R_T[PosMatriz-1]=adc_valor;  
            //USART_Transmit('r');  
        }  
  
        if(Opciones_led==1)  
        {  
            Led_I_T[PosMatriz-1]=adc_valor;  
            //USART_Transmit('I');  
        }  
        /*  
        sprintf (buffer, "%d" , adc_valor );  
        UTransmit_String(buffer);*/  
    }  
}
```

```

void AnalisisNumerico(void)
{
    /*
        for( uint8_t Matriz_Data = 0; Matriz_Data < MuestrasTotales;
Matriz_Data++ )
        {
            sprintf (buffer, "%d" , Led_R_T[Matriz_Data] );
            USART_Transmit('R');
            UTransmit_String(buffer);
        }

        for( uint8_t Matriz_Data = 0; Matriz_Data < MuestrasTotales;
Matriz_Data++ )
        {
            sprintf (buffer, "%d" , Led_I_T[Matriz_Data] );
            USART_Transmit('I');
            UTransmit_String(buffer);
        }
    */

    //GUARDA LOS DATOS DE CADA SEGUNDO EN UNA MATRIZ DE 14
ELEMENTOS
    for( uint8_t dato = 0; dato < DatosxSeg; dato++
){Led_R1s[dato]=Led_R_T[dato];}
    for( uint8_t dato = 0; dato < DatosxSeg; dato++
){Led_R2s[dato]=Led_R_T[dato+14];}
    for( uint8_t dato = 0; dato < DatosxSeg; dato++
){Led_R3s[dato]=Led_R_T[dato+14];}
    for( uint8_t dato = 0; dato < DatosxSeg; dato++
){Led_R4s[dato]=Led_R_T[dato+14];}

```

```

    for( uint8_t dato = 0; dato < DatosxSeg; dato++
){Led_I1s[dato]=Led_I_T[dato];}
    for( uint8_t dato = 0; dato < DatosxSeg; dato++
){Led_I2s[dato]=Led_I_T[dato+14];}
    for( uint8_t dato = 0; dato < DatosxSeg; dato++
){Led_I3s[dato]=Led_I_T[dato+14];}
    for( uint8_t dato = 0; dato < DatosxSeg; dato++
){Led_I4s[dato]=Led_I_T[dato+14];}

    #if (EnviarMatriz == 1)
        UTransmit_String(" TRAMA INIT->->");
        for( uint8_t Matriz_Data = 0; Matriz_Data < DatosxSeg;
Matriz_Data++){sprintf (buffer, "%d" , Led_R1s[Matriz_Data]
);USART_Transmit('r');UTransmit_String(buffer);}
        for( uint8_t Matriz_Data = 0; Matriz_Data < DatosxSeg;
Matriz_Data++){sprintf (buffer, "%d" , Led_R2s[Matriz_Data]
);USART_Transmit('r');UTransmit_String(buffer);}
        for( uint8_t Matriz_Data = 0; Matriz_Data < DatosxSeg;
Matriz_Data++){sprintf (buffer, "%d" , Led_R3s[Matriz_Data]
);USART_Transmit('r');UTransmit_String(buffer);}
        for( uint8_t Matriz_Data = 0; Matriz_Data < DatosxSeg;
Matriz_Data++){sprintf (buffer, "%d" , Led_R4s[Matriz_Data]
);USART_Transmit('r');UTransmit_String(buffer);}
        //-----
        for( uint8_t Matriz_Data = 0; Matriz_Data < DatosxSeg;
Matriz_Data++){sprintf (buffer, "%d" , Led_I1s[Matriz_Data]
);USART_Transmit('i');UTransmit_String(buffer);}
        for( uint8_t Matriz_Data = 0; Matriz_Data < DatosxSeg;
Matriz_Data++){sprintf (buffer, "%d" , Led_I2s[Matriz_Data]
);USART_Transmit('i');UTransmit_String(buffer);}

```

```

        for( uint8_t Matriz_Data = 0; Matriz_Data < DatosxSeg;
Matriz_Data++){sprintf (buffer, "%d" , Led_I3s[Matriz_Data]
);USART_Transmit('i');UTransmit_String(buffer);}
        for( uint8_t Matriz_Data = 0; Matriz_Data < DatosxSeg;
Matriz_Data++){sprintf (buffer, "%d" , Led_I4s[Matriz_Data]
);USART_Transmit('i');UTransmit_String(buffer);}
        UTransmit_String("<-<-FIN TRAMA ");
    #endif

    //CALCULA EL NUMERO MAYOR, MENOR Y LAS VECES QUE SE
REPITE
    USART_Transmit('R');
    CalMayorMenor(Led_R1s);
    CalMayorMenor(Led_R2s);
    CalMayorMenor(Led_R3s);
    CalMayorMenor(Led_R4s);
    USART_Transmit('I');
    CalMayorMenor(Led_I1s);
    CalMayorMenor(Led_I2s);
    CalMayorMenor(Led_I3s);
    CalMayorMenor(Led_I4s);

    //CALCULA EL PROMEDIO DE ROJO E INFRARROJO

    Val_promedio=0;Final_promedio=0;
    for (uint8_t cal_val=0; cal_val < MuestrasTotales; cal_val++)
    {
        Val_promedio = Val_promedio + Led_R_T [cal_val];
    }
    Val_promedio= Val_promedio/MuestrasTotales;
    Final_promedio=Val_promedio;

```

```

    UTransmit_String("PR");sprintf (buffer, "%d" ,
Final_promedio);UTransmit_String(buffer);
    Val_promedio=0;Final_promedio=0;
    for (uint8_t cal_val=0; cal_val < MuestrasTotales; cal_val++)
    {
        Val_promedio = Val_promedio + Led_I_T [cal_val];
    }
    Val_promedio= Val_promedio/MuestrasTotales;
    Final_promedio=Val_promedio;
    UTransmit_String("PI");sprintf (buffer, "%d" ,
Final_promedio);UTransmit_String(buffer);
}
void CalMayorMenor(int Mtr_Mn[])
{
    int mayor, menor;
    uint8_t NumRepMay;
    uint8_t NumRepMen;

    //-----
    //DETERMINO CUAL ES EL MAYOR Y EL MENOR
    //-----
    mayor = Mtr_Mn[0]; //Le asignamos el primer elemento del array
    menor = Mtr_Mn[0]; //Así empezamos a comparar

    for (uint8_t i=0; i<DatosxSeg; i++){
        if (Mtr_Mn[i]> mayor){
            mayor=Mtr_Mn[i];
        }
        if (Mtr_Mn[i]< menor){
            menor=Mtr_Mn[i];
        }
    }
}

```

```

//-----
//DETERMINO CUANTAS VECES SE REPITEN
//-----

    NumRepMay = 0;
    for(uint8_t j = 0; j < DatosxSeg; j++){if ( mayor == Mtr_Mn[j])
{NumRepMay++;}}

    NumRepMen = 0;
    for(uint8_t j = 0; j < DatosxSeg; j++){if ( menor == Mtr_Mn[j])
{NumRepMen++;}}

    UTransmit_String("M");sprintf (buffer, "%d" ,
mayor);UTransmit_String(buffer);
    UTransmit_String("r");sprintf (buffer, "%d" ,
NumRepMay);UTransmit_String(buffer);

    UTransmit_String("m");sprintf (buffer, "%d" ,
menor);UTransmit_String(buffer);
    UTransmit_String("r");sprintf (buffer, "%d" ,
NumRepMen);UTransmit_String(buffer);
}

#if (Adc_resolucion_Op)
    void adc_init_op(void)
    {
        #if (adc_gan10)
            ADMUX=0b00001001;//OKOKOKOKOK //voltaje de
referencia 5 voltios ganancia 10X
        #endif
        ADCSRA=0b00000000;

```

```

        #if (adc_fec4)
            ADCSRA|=(1<<ADEN)|(1<<ADIE)|(1<<ADPS1);
        #endif
        #if (adc_fec8)
ADCSRA|=(1<<ADEN)|(1<<ADIE)|(1<<ADPS1)|(1<<ADPS0);
        #endif
        #if (adc_fec16)
            ADCSRA|=(1<<ADEN)|(1<<ADIE)|(1<<ADPS2);
        #endif
        DIDR0|=(1<<ADC0D)|(1<<ADC1D)|(1<<ADC2D)|(1<<ADC3D)|(1<<ADC
4D)|(1<<ADC5D)|(1<<ADC6D)|(1<<ADC7D);
        sei();//habilita interrupciones globales
    }
    void adc_leer_canal_op(void)
    {
        #if (adc_gan10)
            ADMUX=0b00001001;//OKOKOKOKOK //voltaje de referencia 5
voltios ganancia 10X
        #endif
        _delay_us(130);
        ADCSRA |= (1<<ADSC)|(1<<ADIE);
    }
#endif
ISR(ADC_vect)
{
    _delay_us(130);
    while ((ADCSRA & (1<<ADIF)) == 1);//espera mientras el bit de
interrupcion sea 1
    adc_valor = ADCL;
    /*shift from low level to high level ADC, from 8bit to 10bit*/
    adc_valor += (ADCH<<8);
}

```

ANEXO 3: Programación del formulario 1 (Inicio)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Windows.Forms;
using System.Data;
using System.Data.SqlClient;
using System.Net.Mail;
using System.Net;
using System.Threading.Tasks;

namespace Loggin
{
    public partial class Inicio : System.Web.UI.Page
    {
        string cadena = "workstation id=BDSISTEMA.mssql.somee.com;packet
size=4096;user id=oswaldoarias;pwd=udla2016;data
source=BDSISTEMA.mssql.somee.com;persist security info=False;initial
catalog=BDSISTEMA";
        string contraseña = "";
        public static string apellido = "";
        public static string nombre = "";
        public static string sexo = "";
        public static string nickname = "";
        public static string password = "";
        public static string mail = "";
        string mensaje_validar = "";
        string correo_usuario = "";
        string correo_email = "";
    }
}
```

```

string correo_contraseña = "";
protected void Page_Load(object sender, EventArgs e)
protected void LnkJoggin_Click(object sender, EventArgs e)
{
    this.regresar_inicio();
}
protected void LnkJrear_Click(object sender, EventArgs e)
{
    SqlConnection con_nickname = new SqlConnection(cadena);
    SqlConnection con_mail = new SqlConnection(cadena);
    string nickname = "Select * from Doctores where Nickname =" +
TxtNickname.Text + "";
    string mail = "Select * from Doctores where Mail =" + TxtMail.Text + "";
    SqlCommand comando_nickname = new SqlCommand(nickname,
con_nickname);
    SqlCommand comando_mail = new SqlCommand(mail, con_mail);
    con_nickname.Open();
    con_mail.Open();
    SqlDataReader leer_nickname = comando_nickname.ExecuteReader();
    SqlDataReader leer_mail = comando_mail.ExecuteReader();
    if ((TxtNombre.Text != "") && (TxtApellido.Text != "") &&
(TxtNickname.Text != "") && (DrpSexo.SelectedValue != "Seleccionar...") &&
(TxtMail.Text != "") && (TxtPassword.Text != "") &&
(TxtMail.Text.Contains("@")))
    {
        if (leer_nickname.Read() == true)
            correo_usuario = leer_nickname["Nickname"].ToString();
        if ((correo_usuario == TxtNickname.Text) && (correo_email !=
TxtMail.Text))
        {
            MessageBox.Show("EL NOMBRE DE USUARIO " +
TxtNickname.Text + " YA EXISTE " + "PORFAVOR SELECCIONE OTRO

```

```

NOMBRE DE USUARIO", "ERROR DEL SISTEMA", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
}

if (leer_mail.Read() == true)    {
    correo_email = leer_mail["Mail"].ToString();
    if (((correo_usuario != TxtNickname.Text) && (correo_email ==
TxtMail.Text)) || ((correo_usuario != TxtNickname.Text) && (correo_email ==
TxtMail.Text)))
    {
        MessageBox.Show("EL CORREO " + TxtMail.Text + " ESTE
ASOCIADO A OTRO NOMBRE DE USUARIO " + "PORFAVOR VERIFIQUE
LOS DATOS O SELECCIONE LAS OPCIONES DE RECUPERAR O
GENERAR NUEVA CONTRASEÑA", "ERROR DEL SISTEMA",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

if ((leer_nickname.Read() == false) && (leer_mail.Read() == false))
{
    con_nickname.Close();
    con_mail.Close();
    if ((correo_usuario != TxtNickname.Text) && (correo_email !=
TxtMail.Text))
    {
        SqlConnection con = new SqlConnection(cadena);
        string ad = "insert into Doctores values(@Apellidos, @Nombres,
@Nickname, @Sexo, @Mail, @Password)";
        SqlCommand cmd = new SqlCommand(ad, con);
        con.Open();
    }
}

```

```

cmd.CommandType = CommandType.Text;
cmd.Parameters.Add("@Apellidos", SqlDbType.VarChar,
80).Value = TxtApellido.Text;
cmd.CommandType = CommandType.Text;
cmd.Parameters.Add("@Nombres", SqlDbType.VarChar,
80).Value = TxtNombre.Text;
cmd.CommandType = CommandType.Text;
cmd.Parameters.Add("@Nickname", SqlDbType.VarChar,
30).Value = TxtNickname.Text;
cmd.CommandType = CommandType.Text;
cmd.Parameters.Add("@Sexo", SqlDbType.VarChar, 50).Value
= DrpSexo.SelectedValue;
cmd.CommandType = CommandType.Text;
cmd.Parameters.Add("@Mail", SqlDbType.VarChar, 50).Value =
TxtMail.Text;
cmd.CommandType = CommandType.Text;
cmd.Parameters.Add("@Password", SqlDbType.VarChar,
20).Value = TxtPassword.Text;
cmd.ExecuteNonQuery();
con.Close();
string contraseña_remitente = "Oswaldo1234";
string mensaje_remitente = string.Empty;
string destinatario = TxtMail.Text;
string remitente = "sistema_control@hotmail.com";
string asunto = "VALIDACIÓN DE CUENTA";
if (DrpSexo.SelectedValue == "Mujer")
{
    mensaje_validar = "BIENVENIDA DOCTORA";
}
if (DrpSexo.SelectedValue == "Hombre")
{
    mensaje_validar = "BIENVENIDA DOCTOR";
}

```

```

    }
    if (DrpSexo.SelectedValue == "Sin especificar")
    {
        mensaje_validar = "BIENVENIDA DOCTORA/DOCTOR";
    }

    string cuerpo_mensaje = mensaje_validar + " " +
    TxtNombre.Text + " " + TxtApellido.Text + " AL SISTEMA DE MONITOREO Y
    CONTROL, RECUERDE QUE SUS DATOS PARA EL INCIO DE SESION
    SON: USUARIO: " + TxtNickname.Text + " CONTRASEÑA: " +
    TxtPassword.Text;

    MailMessage ms = new MailMessage(remitente, destinatario,
    asunto, cuerpo_mensaje);

    SmtplibClient smtp = new SmtplibClient("smtp.live.com", 587);
    smtp.EnableSsl = true;
    smtp.Credentials = new
    NetworkCredential("sistema_control@hotmail.com", contraseña_remitente);
    try
    {
        Task.Run(() =>
        {
            smtp.Send(ms);
            ms.Dispose();
            this.regresar_inicio();
        }
    );
    MessageBox.Show("SE ENVIÓ UN MENSAJE DE
    VALIDACIÓN A LA SIGUIENTE DE DIRECCION DE CORREO
    ELECTRÓNICO: " + TxtMail.Text, "NOTIFICACION",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
    this.regresar_inicio();
}

```

```
        catch (Exception ex)
        {
            MessageBox.Show("ERROR AL ENVIAR EL CORREO
ELECTRÓNICO" + ex.Message);
        }
    }
}
else
{
    MessageBox.Show("VERIFICAR EL INGRESO DE LOS DATOS,
RECUERDE QUE TODOS LOS CAMPOS DEBEN SER INGRESADOS Y EL
CORREO DEBE CONSTAR CON EL CARACTER @ ", "ERROR DEL
SISTEMA", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

}

protected void DrpSexo_SelectedIndexChanged(object sender, EventArgs
e)
{
}

protected void DrpSexo_CreatingModelDataSource(object sender,
CreatingModelDataSourceEventArgs e)
{
}

protected void LnkJngresar_Click1(object sender, EventArgs e)
{
```

```

Principal.cambiar = 1;
if ((TxtUsuario.Text == "") && (TxtContraseña.Text == ""))
{
    Response.Write("<script language=javascript>");
    Response.Write("alert('PORFAVOR INGRESE NOMBRE DE
USUARIO Y CONTRASEÑA')");
    Response.Write("</script>");
}
if ((TxtUsuario.Text != "") && (TxtContraseña.Text == ""))
{
    Response.Write("<script language=javascript>");
    Response.Write("alert('PORFAVOR INGRESE NOMBRE DE
USUARIO Y CONTRASEÑA')");
    Response.Write("</script>");
}
if ((TxtUsuario.Text == "") && (TxtContraseña.Text != ""))
{
    Response.Write("<script language=javascript>");
    Response.Write("alert('PORFAVOR INGRESE NOMBRE DE
USUARIO Y CONTRASEÑA')");
    Response.Write("</script>");
}
if ((TxtUsuario.Text != "") && (TxtContraseña.Text != ""))
{
    SqlConnection con_usuario = new SqlConnection(cadena);
    string usuario = "Select * from Doctores where Nickname =" +
TxtUsuario.Text + """;
    SqlCommand comando_usuario = new SqlCommand(usuario,
con_usuario);
    con_usuario.Open();
    SqlDataReader leer_usuario = comando_usuario.ExecuteReader();
    if (leer_usuario.Read() == true)

```

```
{
    contraseña = leer_usuario["Password"].ToString();
    if (contraseña == TxtContraseña.Text)
    {
        Inicio.apellido = leer_usuario["Apellidos"].ToString();
        Inicio.nombre = leer_usuario["Nombres"].ToString();
        Inicio.sexo = leer_usuario["Sexo"].ToString();
        Inicio.nickname = leer_usuario["Nickname"].ToString();
        Inicio.password = leer_usuario["Password"].ToString();
        Inicio.mail = leer_usuario["Mail"].ToString();
        con_usuario.Close();
        Response.Redirect("Presentación.aspx?parametro=");
    }
    else
    {
        con_usuario.Close();
        MessageBox.Show("USUARIO NO EXISTE VERIFIQUE LOS
DATOS INGRESADOS O REGISTRARSE PARA INGRESAR AL SISTEMA",
"ERROR DEL SISTEMA", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
else
{
    con_usuario.Close();
    MessageBox.Show("USUARIO NO EXISTE VERIFIQUE LOS
DATOS INGRESADOS O REGISTRARSE PARA INGRESAR AL SISTEMA",
"ERROR DEL SISTEMA", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
}
```

```
protected void LnkJRegistro_Click(object sender, EventArgs e)
{
    LblInicio.Text = "CREACIÓN DE USUARIO";
    LblUsuario.Visible = (false);
    TxtUsuario.Visible = (false);
    LblContraseña.Visible = (false);
    TxtContraseña.Visible = (false);
    LnkJIngresar.Visible = (false);
    LnkJRegistro.Visible = (false);
    LnkJRecordar.Visible = (false);

    LblNombre.Visible = (true);
    TxtNombre.Visible = (true);
    LblApellidos.Visible = (true);
    TxtApellido.Visible = (true);
    LblNickname.Visible = (true);
    TxtNickname.Visible = (true);
    LblSexo.Visible = (true);
    DrpSexo.Visible = (true);
    LblMail.Visible = (true);
    TxtMail.Visible = (true);
    LblPassword.Visible = (true);
    TxtPassword.Visible = (true);
    LnkJLoggiin.Visible = (true);
    LnkJCrear.Visible = (true);
    TxtNombre.Text = "";
    TxtApellido.Text = "";
    TxtNickname.Text = "";
    TxtMail.Text = "";
    DrpSexo.SelectedValue = "Seleccionar...";
    TxtPassword.Text = "";
```

```
}  
public void regresar_inicio ()  
{  
    LblInicio.Text = "INICIO DE SESIÓN";  
    LblUsuario.Visible = (true);  
    TxtUsuario.Visible = (true);  
    LblContraseña.Visible = (true);  
    TxtContraseña.Visible = (true);  
    LnkJngresar.Visible = (true);  
    LnkJregistro.Visible = (true);  
    LnkJrecordar.Visible = (true);  
    LblNombre.Visible = (false);  
    TxtNombre.Visible = (false);  
    LblApellidos.Visible = (false);  
    TxtApellido.Visible = (false);  
    LblNickname.Visible = (false);  
    TxtNickname.Visible = (false);  
    LblSexo.Visible = (false);  
    DrpSexo.Visible = (false);  
    LblMail.Visible = (false);  
    TxtMail.Visible = (false);  
    LblPassword.Visible = (false);  
    TxtPassword.Visible = (false);  
    LnkJloggiin.Visible = (false);  
    LnkJcrear.Visible = (false);  
    TxtUsuario.Text = "";  
    TxtContraseña.Text = "";  
}  
protected void BtnRecuperar_contraseña_Click(object sender, EventArgs e)  
{  
    if ((TxtUsuario_recuperar.Text == "") &&  
(TxtEmail_recuperar.Text == ""))
```

```

{
    MessageBox.Show("INGRESE NOMBRE DE USUARIO O CORREO
ELECTRONICO PARA REALIZAR LA RECUPERACIÓN DE SU
CONTRASEÑA", "ERROR DEL SISTEMA", MessageBoxButtons.OK,
MessageBoxIcon.Error);
}
if ((TxtUsuario_recuperar.Text != "") && (TxtEmail_recuperar.Text ==
""))
{
    SqlConnection con_recuperar_usuario = new
SqlConnection(cadena);
    string recuperar_usuario = "Select * from Doctores where Nickname
=" + TxtUsuario_recuperar.Text + "";
    SqlCommand comando_recuperar_usuario = new
SqlCommand(recuperar_usuario, con_recuperar_usuario);
    con_recuperar_usuario.Open();
    SqlDataReader leer_recuperar_usuario =
comando_recuperar_usuario.ExecuteReader();
    if (leer_recuperar_usuario.Read() == true)
    {
        correo_email = leer_recuperar_usuario["Mail"].ToString();
        correo_usuario = leer_recuperar_usuario["Nickname"].ToString();
        apellido = leer_recuperar_usuario["Apellidos"].ToString();
        nombre = leer_recuperar_usuario["Nombres"].ToString();
        sexo = leer_recuperar_usuario["Sexo"].ToString();
        correo_contraseña =
leer_recuperar_usuario["Password"].ToString();
        string contraseña_remitente = "Oswaldo1234";
        string mensaje_remitente = string.Empty;
        string destinatario = correo_email;
        string remitente = "sistema_control@hotmail.com";
    }
}

```

```

        string asunto = "DATOS PERSONALES DEL SISTEMA DE
MONITOREO Y CONTROL";
        if (sexo == "Mujer")
        {
            mensaje_validar = "SALUDOS DOCTORA";
        }
        if (sexo == "Hombre")
        {
            mensaje_validar = "SALUDOS DOCTOR";
        }
        if (sexo == "Sin especificar")
        {
            mensaje_validar = "SALUDOS DOCTORA/DOCTOR";
        }

        string cuerpo_mensaje = mensaje_validar + " " + nombre + " " +
apellido + " ESTE ES UN MENSAJE AUTOMÁTICO PARA RECORDARLE SU
USUARIO Y CONTRASEÑA USUARIO: " + correo_usuario + "
CONTRASEÑA: " + correo_contraseña;

        MailMessage ms = new MailMessage(remitente, destinatario,
asunto, cuerpo_mensaje);
        SmtplibClient smtp = new SmtplibClient("smtp.live.com", 587);
        smtp.EnableSsl = true;
        smtp.Credentials = new
NetworkCredential("sistema_control@hotmail.com", contraseña_remitente);
        try
        {
            Task.Run(() =>
            {
                smtp.Send(ms);
                ms.Dispose();
                this.regresar_inicio();
            });
        }
    }
}

```

```

    }
    );
    MessageBox.Show("SE ENVIO SU NOMBRE DE USUARIO Y
CONTRASEÑA AL SIGUIENTE CORREO: " + correo_email,
"NOTIFICACION", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show("ERROR AL ENVIAR EL CORREO
ELECTRÓNICO" + ex.Message);
    }
    PanelPrincipal.Visible = (true);
    PanelRecuperar.Visible = (false);
    TxtUsuario_recuperar.Text = "";
    TxtEmail_recuperar.Text = "";
    }
    else
    {
        MessageBox.Show("DATOS NO EXISTEN", "ERROR DEL
SISTEMA", MessageBoxButtons.OK, MessageBoxIcon.Error);
        TxtUsuario_recuperar.Text = "";
        TxtEmail_recuperar.Text = "";
    }
    con_recuperar_usuario.Close();
    }
    if ((TxtUsuario_recuperar.Text == "") && (TxtEmail_recuperar.Text != "")
|| ((TxtUsuario_recuperar.Text != "") && (TxtEmail_recuperar.Text != "")))
    {
        if(TxtEmail_recuperar.Text.Contains("@"))
        {
            SqlConnection con_recuperar_email = new
SqlConnection(cadena);

```

```

        string recuperar_email = "Select * from Doctores where Mail =" +
TxtEmail_recuperar.Text + """;
        SqlCommand comando_recuperar_email = new
SqlCommand(recuperar_email, con_recuperar_email);
        con_recuperar_email.Open();
        SqlDataReader leer_recuperar_email =
comando_recuperar_email.ExecuteReader();
        if (leer_recuperar_email.Read() == true)
        {
            correo_email = leer_recuperar_email["Mail"].ToString();
            correo_usuario = leer_recuperar_email["Nickname"].ToString();
            apellido = leer_recuperar_email["Apellidos"].ToString();
            nombre = leer_recuperar_email["Nombres"].ToString();
            sexo = leer_recuperar_email["Sexo"].ToString();
            correo_contraseña =
MONITOREO Y CONTROL";
            if (sexo == "Mujer")
            {
                mensaje_validar = "SALUDOS DOCTORA";
            }
            if (sexo == "Hombre")
            {
                mensaje_validar = "SALUDOS DOCTOR";
            }
            if (sexo == "Sin especificar")
            {
                mensaje_validar = "SALUDOS DOCTORA/DOCTOR";
            }
            string cuerpo_mensaje = mensaje_validar + " " + nombre + " " +
apellido + " ESTE ES UN MENSAJE AUTOMÁTICO PARA RECORDARLE SU
USUARIO Y CONTRASEÑA USUARIO: " + correo_usuario + "
CONTRASEÑA: " + correo_contraseña;

```

```

        MailMessage ms = new MailMessage(remitente, destinatario,
asunto, cuerpo_mensaje);
        SmtplibClient smtp = new SmtplibClient("smtp.live.com", 587);
        smtp.EnableSsl = true;
        smtp.Credentials = new
NetworkCredential("sistema_control@hotmail.com", contraseña_remitente);
        try
        {
            Task.Run(() =>
            {
                smtp.Send(ms);
                ms.Dispose();
                this.regresar_inicio();
            }
        );
            MessageBox.Show("SE ENVIO SU NOMBRE DE USUARIO
Y CONTRASEÑA AL SIGUIENTE CORREO: " + correo_email,
"NOTIFICACION", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        catch (Exception ex)
        {
            MessageBox.Show("ERROR AL ENVIAR EL CORREO
ELECTRÓNICO" + ex.Message);
        }
        PanelPrincipal.Visible = (true);
        PanelRecuperar.Visible = (false);
        TxtUsuario_recuperar.Text = "";
        TxtEmail_recuperar.Text = "";
    }
    else
    {

```

```

        MessageBox.Show("DATOS NO EXISTEN VUELVE INGRESAR
O REGISTRESE PARA PODER INICIAR SESIÓN", "ERROR DEL SISTEMA",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        TxtUsuario_recuperar.Text = "";
        TxtEmail_recuperar.Text = "";
    }
    con_recuperar_email.Close();
}
else
{
    MessageBox.Show("RECUERDE QUE EL CORREO DEBE
TENER EL CARACTER '@'", "ERROR DEL SISTEMA",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    TxtUsuario_recuperar.Text = "";
    TxtEmail_recuperar.Text = "";
}
}
}
protected void BtnRegresar_Click(object sender, EventArgs e)
{
    PanelPrincipal.Visible = (true);
    PanelRecuperar.Visible = (false);
    TxtUsuario_recuperar.Text = "";
    TxtEmail_recuperar.Text = "";
}
protected void LnkRecordar_Click(object sender, EventArgs e)
{
    PanelPrincipal.Visible = (false);
    PanelRecuperar.Visible = (true);
}
}
}
}

```

ANEXO 4: Programación del formulario 2 (Presentación)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Loggin
{
    public partial class Prsentación : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Inicio.sexo=="Mujer")
            {
                LblSaludo1.Text= "BIENVENIDA DOCTORA";
                LblSaludo2.Text = Inicio.nombre + " " + Inicio.apellido;
            }
            if (Inicio.sexo == "Hombre")
            {
                LblSaludo1.Text = "BIENVENIDO DOCTOR";
                LblSaludo2.Text = Inicio.nombre + " " + Inicio.apellido;
            }
            if (Inicio.sexo == "Sin especificar")
            {
                LblSaludo1.Text = "BIENVENIDA DOCTOR/DOCTORA";
                LblSaludo2.Text = Inicio.nombre + " " + Inicio.apellido;
            }
        }
    }
}

```

ANEXO 5: Programación del formulario 3 (Ingresar_Paciente)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Windows.Forms;
using System.Data;
using System.Data.SqlClient;
using System.Net.Mail;
using System.Net;
using System.Threading.Tasks;

namespace Loggin
{
    public partial class Ingresar_paciente : System.Web.UI.Page
    {
        string cadena = "workstation id=BDSISTEMA.mssql.somee.com;packet
size=4096;user id=oswaldoarias;pwd=udla2016;data
source=BDSISTEMA.mssql.somee.com;persist security info=False;initial
catalog=BDSISTEMA";
        string genero = "";
        string verificar_nickname="";    string codigo_verificacion = "";
        string validar_genero = "";
        string mensaje_validar = "";
        DateTime fecha;
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Inicio.sexo == "Mujer")
            {

```

```

        genero = "DE LA DOCTORA";
    }
    if (Inicio.sexo == "Hombre")
    {
        genero = "DEL DOCTOR";
    }
    if (Inicio.sexo == "Sin especificar")
    {
        genero = "DOCTOR/DOCTORA";
    }
    LblIngreso.Text = "INGRESO DE PACIENTES " + genero + " " +
Inicio.nombre + " " + Inicio.apellido;
    DrpDia.Items.Add(new ListItem("Día", "Valor"));
    DrpAño.Items.Add(new ListItem("Año", "Valor"));
    for (int i = 1; i <= 31; i++)
    {
        DrpDia.Items.Add(i.ToString());
    }
    for (int i = DateTime.Today.Year; i >= 1900; i--)
    {
        DrpAño.Items.Add(i.ToString());
    }
    verificar_nickname = Inicio.nickname;
}

protected void BtnCrear_paciente_Click(object sender, EventArgs e)
{
    string ingreso_fecha = DrpMes.SelectedValue + "/" +
DrpDia.SelectedValue + "/" + DrpAño.SelectedValue;
    SqlConnection con_verificar = new SqlConnection(cadena);
    string verificar = "Select * from Ingreso_Paciente where Verificación ="
+ Inicio.nickname + TxtCedula.Text + """;
}

```

```

        SqlCommand comando_verificar = new SqlCommand(verificar,
con_verificar);
        con_verificar.Open();
        SqlDataReader leer_verificar = comando_verificar.ExecuteReader();
        if ((TxtApellidos.Text != "") && (TxtNombres.Text != "") &&
(TxtCedula.Text != "") && (DrpMes.SelectedValue != "Mes") &&
(DrpDia.SelectedValue != "Día") && (DrpMes.SelectedValue != "Año") &&
(DrpMes.SelectedValue != "Seleccionar") && (TxtMail.Text != "") &&
(TxtTelefono.Text != "") && (TxtDiagnostico.Text != "") &&
(TxtMail.Text.Contains("@")))
        {
            fecha = Convert.ToDateTime(ingreso_fecha);
            if (leer_verificar.Read()==true)
            {
                codigo_verificacion = leer_verificar["Verificación"].ToString();
                if (codigo_verificacion==Inicio.nickname + TxtCedula.Text)
                {
                    MessageBox.Show("PACIENTE YA EXISTE PORFAVOR
VERIFIQUE LOS DATOS", "ERROR DEL SISTEMA", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                }
                if (codigo_verificacion != Inicio.nickname + TxtCedula.Text)
                {
                    SqlConnection con = new SqlConnection(cadena);
                    string ad = "insert into Ingreso_Paciente
values(@Nickname_Médico, @Apellidos, @Nombres, @Cédula, @Sexo,
@Fecha_nacimiento, @Mail, @Teléfono, @Diagnóstico, @Verificación)";
                    SqlCommand cmd = new SqlCommand(ad, con);
                    con.Open();
                    cmd.CommandType = CommandType.Text;
                    cmd.Parameters.Add("@Nickname_Médico",
SqlDbType.VarChar, 30).Value = Inicio.nickname;

```

```

        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Apellidos", SqlDbType.VarChar,
80).Value = TxtApellidos.Text;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Nombres", SqlDbType.VarChar,
80).Value = TxtNombres.Text;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Cédula", SqlDbType.VarChar,
12).Value = TxtCedula.Text;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Sexo", SqlDbType.VarChar, 10).Value
= DrpSexo.SelectedValue;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Fecha_nacimiento",
SqlDbType.Date).Value = fecha;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Mail", SqlDbType.VarChar, 30).Value =
TxtMail.Text;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Teléfono", SqlDbType.VarChar,
50).Value = TxtTelefono.Text;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Diagnóstico", SqlDbType.VarChar,
300).Value = TxtDiagnostico.Text;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Verificación", SqlDbType.VarChar,
100).Value = Inicio.nickname + TxtCedula.Text;
        cmd.ExecuteNonQuery();
        con.Close();
        MessageBox.Show("EL PACIENTE HA SIDO INGRESADO
CORRECTAMENTE ", "NOTIFICACION", MessageBoxButtons.OK,
MessageBoxIcon.Information);

```

```

        this.limpiar();
    }
}
if (leer_verificar.Read()==false)
{
    fecha = Convert.ToDateTime(ingreso_fecha);
    if (codigo_verificacion != Inicio.nickname + TxtCedula.Text)
    {
        con_verificar.Close();
        SqlConnection con = new SqlConnection(cadena);
        string ad = "insert into Ingreso_Paciente
values(@Nickname_Médico, @Apellidos, @Nombres, @Cédula, @Sexo,
@Fecha_nacimiento, @Mail, @Teléfono, @Diagnóstico, @Verificación)";
        SqlCommand cmd = new SqlCommand(ad, con);
        con.Open();
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Nickname_Médico",
SqlDbType.VarChar, 30).Value = Inicio.nickname;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Apellidos", SqlDbType.VarChar,
80).Value = TxtApellidos.Text;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Nombres", SqlDbType.VarChar,
80).Value = TxtNombres.Text;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Cédula", SqlDbType.VarChar,
12).Value = TxtCedula.Text;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Sexo", SqlDbType.VarChar, 10).Value
= DrpSexo.SelectedValue;
        cmd.CommandType = CommandType.Text;

```

```

        cmd.Parameters.Add("@Fecha_nacimiento",
SqlDbType.Date).Value = fecha;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Mail", SqlDbType.VarChar, 30).Value =
TxtMail.Text;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Teléfono", SqlDbType.VarChar,
50).Value = TxtTelefono.Text;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Diagnóstico", SqlDbType.VarChar,
300).Value = TxtDiagnostico.Text;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Verificación", SqlDbType.VarChar,
100).Value = Inicio.nickname + TxtCedula.Text;
        cmd.ExecuteNonQuery();
        con.Close();
        string contraseña_remitente = "Oswaldo1234";
        string mensaje_remitente = string.Empty;
        string destinatario = TxtMail.Text;
        string remitente = "sistema_control@hotmail.com";
        string asunto = "MENSAJE DE BIENVENIDA";
        if (DrpSexo.SelectedValue == "Femenino")
        {
            validar_genero = "BIENVENIDA";
        }
        if (DrpSexo.SelectedValue == "Masculino")
        {
            validar_genero = "BIENVENIDO";
        }
        if (DrpSexo.SelectedValue == "Sin especificar..")
        {
            validar_genero = "BIENVENIDA/BIENVENIDO";
        }

```

```

    }
    if (Inicio.sexo == "Mujer")
    {
        mensaje_validar = "DE LA DOCTORA";
    }
    if (Inicio.sexo == "Hombre")
    {
        mensaje_validar = "DEL DOCTOR";
    }
    if (Inicio.sexo == "Sin especificar")
    {
        mensaje_validar = "DE LA DOCTORA/DOCTOR";
    }
    string cuerpo_mensaje = validar_genero + " " +
    TxtNombres.Text + " " + TxtApellidos.Text + " AL SISTEMA DE MONITOREO
    Y CONTROL " + mensaje_validar + " " + Inicio.nombre + " " + Inicio.apellido;
    MailMessage ms = new MailMessage(remitente, destinatario,
    asunto, cuerpo_mensaje);
    SmtplibClient smtp = new SmtplibClient("smtp.live.com", 587);
    smtp.EnableSsl = true;
    smtp.Credentials = new
    NetworkCredential("sistema_control@hotmail.com", contraseña_remitente);
    try
    {
        Task.Run(() =>
        {
            smtp.Send(ms);
            ms.Dispose();
        }
    );

```

```

        MessageBox.Show("EL PACIENTE HA SIDO INGRESADO
CORRECTAMENTE ", "NOTIFICACION", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show("ERROR AL ENVIAR EL CORREO
ELECTRÓNICO" + ex.Message);
    }

    this.limpiar();
}
}
}
else
{
    MessageBox.Show("VERIFICAR EL INGRESO DE LOS DATOS,
RECUERDE QUE TODOS LOS CAMPOS DEBEN SER INGRESADOS Y EL
CORREO DEBE CONSTAR CON EL CARACTER @ ", "ERROR DEL
SISTEMA", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
protected void DrpMes_SelectedIndexChanged(object sender, EventArgs
e)
{
}

protected void DrpDia_SelectedIndexChanged(object sender, EventArgs
e)

```

```

{

}
public void limpiar()
{
    TxtApellidos.Text = "";
    TxtNombres.Text = "";
    TxtCedula.Text = "";
    DrpMes.Items.Add(new ListItem("Mes", "Valor"));
    DrpDia.Items.Add(new ListItem("Día", "Valor"));
    DrpAño.Items.Add(new ListItem("Año", "Valor"));
    DrpSexo.Items.Add(new ListItem("Sexo", "Valor"));
    TxtMail.Text = "";
    TxtTelefono.Text = "";
    TxtDiagnostico.Text = "";
}
}

```

ANEXO 6: Programación del formulario 4 (Control_Paciente)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Windows.Forms;
using System.Data;
using System.Data.SqlClient;
using System.Net.Mail;
using System.Net;
using System.Threading.Tasks;

```

```

namespace Loggin
{
    public partial class Controlar_paciente : System.Web.UI.Page
    {
        string cadena = "workstation id=BDSISTEMA.mssql.somee.com;packet
size=4096;user id=oswaldoarias;pwd=udla2016;data
source=BDSISTEMA.mssql.somee.com;persist security info=False;initial
catalog=BDSISTEMA";
        string genero = "";
        string mail = "";
        string telefono = "";
        DataSet resultados = new DataSet();
        DataView mifiltro;
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Inicio.sexo == "Mujer")
            {
                genero = "DE LA DOCTORA";
            }
            if (Inicio.sexo == "Hombre")
            {
                genero = "DEL DOCTOR";
            }
            if (Inicio.sexo == "Sin especificar")
            {
                genero = "DOCTOR/DOCTORA";
            }
            LblBienvenida.Text = "CONTROL DE PACIENTES " + genero + " " +
Inicio.nombre + " " + Inicio.apellido;
            string querypacientes = "SELECT * FROM [Control_Paciente] where
Nickname_M = " + Inicio.nickname + """;
            Listado_Pacientes(querypacientes);
        }
    }
}

```

```

}
protected void CheCedula_CheckedChanged(object sender, EventArgs e)
{
    CheNombre.Checked = false;
    TxtConsulta.Text = "";
    TxtConsulta.Enabled = true;
}
protected void CheNombre_CheckedChanged(object sender, EventArgs e)
{
    CheCedula.Checked = false;
    TxtConsulta.Text = "";
    TxtConsulta.Enabled = true;
}
protected void BtnGuardar_Click(object sender, EventArgs e)
{
    if (TxtObservacion.Text != "")
    {
        Modificar_Paciente(idPaciente.Text, TxtObservacion.Text);
    }
    if (TxtObservacion.Text == "")
    {
        MessageBox.Show("EL CAMPO OBSERVACION NO PUEDE
ESTAR VACIO", "ERROR DEL SISTEMA", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
}
public void Listado_Pacientes(string query)
{
    try
    {
        using (SqlConnection connection = new SqlConnection(cadena))
        {

```

```

        SqlCommand command = new SqlCommand(query, connection);
        connection.Open();
        this.GridView1.DataSource = command.ExecuteReader();
        this.GridView1.DataBind();
        connection.Close();
    }
}
catch (Exception ex)
{
    MessageBox.Show("Ha ocurrido un Error al intentar mostrar
informacion de los Pacientes.");
}
}
public void Modificar_Paciente(string id,string observacion)
{
    try
    {
        string query = "update Control_Paciente set Observación = " +
observacion + " where idPaciente=" + id + """;
        using (SqlConnection connection = new SqlConnection(cadena))
        {
            SqlCommand command = new SqlCommand(query, connection);
            connection.Open();
            command.ExecuteReader();
            string querypacientes = "SELECT * FROM [Control_Paciente]
where Nickname_M =" + Inicio.nickname + """;
            Listado_Pacientes(querypacientes);
            connection.Close();
        }
    }
}
catch (Exception ex)
{

```

```

        MessageBox.Show("Ha ocurrido un Error al intentar modificar al
Paciente.");
    }
    return;
}
protected void buscar_Click(object sender, EventArgs e)
{
    string querypacientes = "";
    if (TxtConsulta.Text != "")
    {
        if (CheNombre.Checked)
        {
            querypacientes = "SELECT * FROM [Control_Paciente] where
Nickname_M='" + Inicio.nickname + "' and Paciente like '%" + TxtConsulta.Text
+ "%'";
        }
        if (CheCedula.Checked)
        {
            querypacientes = "SELECT * FROM [Control_Paciente] where
Nickname_M='" + Inicio.nickname + "' and Cédula_P like '%" +
TxtConsulta.Text + "%'";
        }
        Listado_Pacientes(querypacientes);
    }
    else
    {
        querypacientes = "SELECT * FROM [Control_Paciente] where
Nickname_M='" + Inicio.nickname + "'";
        Listado_Pacientes(querypacientes);
    }
}
}

```

```

protected void GridView1_SelectedIndexChanged1(object sender,
EventArgs e)
{
    TxtCedula.Text = GridView1.SelectedRow.Cells[2].Text;
    TxtNombre.Text = GridView1.SelectedRow.Cells[3].Text;
    TxtFecha.Text = GridView1.SelectedRow.Cells[4].Text;
    TxtOxigeno.Text = GridView1.SelectedRow.Cells[5].Text;
    TxtPulso.Text = GridView1.SelectedRow.Cells[6].Text;
    TxtObservacion.Text = GridView1.SelectedRow.Cells[7].Text;
    idPaciente.Text = GridView1.SelectedRow.Cells[8].Text;
    SqlConnection con_cedula = new SqlConnection(cadena);
    string ingreso_cedula = "Select * from Ingreso_Paciente where Cédula
=" + TxtCedula.Text + """;
    SqlCommand comando_cedula = new SqlCommand(ingreso_cedula,
con_cedula);
    con_cedula.Open();
    SqlDataReader leer_cedula = comando_cedula.ExecuteReader();
    if (leer_cedula.Read() == true)
    {
        mail = leer_cedula["Mail"].ToString();
        telefono = leer_cedula["Teléfono"].ToString();
    }
    TxtMail.Text = mail;
    TxtTelefono.Text = telefono;
    TxtObservacion.Enabled = true;
    BtnGuardar.Enabled = true;
    con_cedula.Close();
}

protected void BtnEliminar_Click(object sender, EventArgs e)
{
    SqlConnection cnx;

```

```

SqlCommand query;
int res;
try{
    cnx = new SqlConnection(cadena);
    cnx.Open();
    query = new SqlCommand();
    query.CommandType = CommandType.Text;
    query.CommandText = "Delete from Ingreso_Paciente Where
Nickname_Médico =" + TxtEliminar.Text + """;
    query.Connection = cnx;
    res = query.ExecuteNonQuery();
    if (res != 0){
        Response.Write("<script language=javascript>");
        Response.Write("alert('Alumno Eliminado')");
        Response.Write("</script>");
    } else {
        Response.Write("<script language=javascript>");
        Response.Write("alert('El Alumno no Existe')");
        Response.Write("</script>");
    }
    cnx.Close();

} catch (Exception ex) {
    Response.Write("" + ex.Message);
}
}
}
}

```

ANEXO 7: Programación de la opción “Cambiar_Contraseña”

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Windows.Forms;
using System.Data;
using System.Data.SqlClient;
using System.Net.Mail;
using System.Net;
using System.Threading.Tasks;

namespace Loggin
{
    public partial class Cambiar_contraseña : System.Web.UI.Page
    {
        string cadena = "workstation id=BDSISTEMA.mssql.somee.com;packet
size=4096;user id=oswaldoarias;pwd=udla2016;data
source=BDSISTEMA.mssql.somee.com;persist security info=False;initial
catalog=BDSISTEMA";
        string correo_email = "";
        string mensaje_validar = "";
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Inicio.sexo == "Mujer")
            {
                LblAcronimo.Text = "Doctora";
                LblIdentificacion.Text = Inicio.nombre + " " + Inicio.apellido;
            }
            if (Inicio.sexo == "Hombre")

```

```
{
    LblAcronimo.Text = "Doctor";
    LblIdentificacion.Text = Inicio.nombre + " " + Inicio.apellido;
}
if (Inicio.sexo == "Sin especificar")
{
    LblAcronimo.Text = "Doctor/Doctora";
    LblIdentificacion.Text = Inicio.nombre + " " + Inicio.apellido;
}
}

protected void Button1_Click(object sender, EventArgs e)
{
}

protected void BtnCancelar_Click(object sender, EventArgs e)
{
    if (Principal.cambiar==1)
    {
        Response.Redirect("Presentación.aspx?parametro=");
    }
    if (Principal.cambiar == 2)
    {
        Response.Redirect("Ingresar_paciente.aspx?parametro=");
    }
    if (Principal.cambiar == 3)
    {
        Response.Redirect("Controlar_paciente.aspx?parametro=");
    }
}
```

```

protected void BtnCambiar_Click(object sender, EventArgs e)
{
    if (TxtRecuperar.Text == "" || TxtConfirmar.Text == "")
    {
        MessageBox.Show("PORFAVOR INGRESE LOS DATOS EN LOS
        ESPACIOS SOILICITADOS ", "ERROR DEL SISTEMA",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    if (TxtRecuperar.Text != "" && TxtConfirmar.Text != "" &&
    TxtRecuperar.Text != Inicio.password)
    {
        if (TxtRecuperar.Text == TxtConfirmar.Text)
        {
            SqlConnection con = new SqlConnection(cadena);
            String ad = "update Doctores set Password = " +
            TxtRecuperar.Text + " where Nickname = " + Inicio.nickname + "";";
            SqlCommand cmd = new SqlCommand(ad, con);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            string contraseña_remitente = "Oswaldo1234";
            string mensaje_remitente = string.Empty;
            string destinatario = Inicio.mail;
            string remitente = "sistema_control@hotmail.com";
            string asunto = "CAMBIO DE CONTRASEÑA";
            if (Inicio.sexo == "Mujer")
            {
                mensaje_validar = "SALUDOS DOCTORA";
            }
            if (Inicio.sexo == "Hombre")
            {

```

```

        mensaje_validar = "SALUDOS DOCTOR";
    }
    if (Inicio.sexo == "Sin especificar")
    {
        mensaje_validar = "SALUDOS DOCTORA/DOCTOR";
    }
    string cuerpo_mensaje = mensaje_validar + " mensaje_validar" +
Inicio.nombre + " " + Inicio.apellido + " SU CLAVE HA SIDO CAMBIAD
RECUERDE QUE SUS DATOS AHORA SON LOS SIGUIENTES : USUARIO:
" + Inicio.nickname + " CONTRASEÑA: " + TxtRecuperar.Text;
    MailMessage ms = new MailMessage(remiteante, destinatario,
asunto, cuerpo_mensaje);
    SmtClient smtp = new SmtClient("smtp.live.com", 587);
    smtp.EnableSsl = true;
    smtp.Credentials = new
NetworkCredential("sistema_control@hotmail.com", contraseña_remitente);
    try
    {
        Task.Run(() =>
        {
            smtp.Send(ms);
            ms.Dispose();
        }
    );
    MessageBox.Show("LA CONTRASEÑA HA SIDO CAMBIADA,
SE ENVIO SU NOMBRE DE USUARIO Y NUEVA CONTRASEÑA AL
SIGUIENTE CORREO: " + Inicio.mail + " SERA REDIRIGIDO A LA
PANTALLA DE LOGIN PARA EL INICIO DE SESION", "NOTIFICACION",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {

```

```
        MessageBox.Show("ERROR AL ENVIAR EL CORREO  
ELECTRÓNICO" + ex.Message);  
    }  
    Response.Redirect("Inicio.aspx?parametro=");  
}  
if (TxtRecuperar.Text != TxtConfirmar.Text)  
{  
    MessageBox.Show("LAS CONTRASEÑAS NO COINCIDEN  
PORFAVOR VUELVA A INGRESAR", "ERROR DEL SISTEMA",  
MessageBoxButtons.OK, MessageBoxIcon.Error);  
    TxtRecuperar.Text = "";  
    TxtConfirmar.Text = "";  
}  
}  
if (TxtRecuperar.Text == Inicio.password)  
{  
    MessageBox.Show("CONTRASEÑA YA EXISTE PORFAVOR  
INGRESE UNA NUEVA CONTRASEÑA ", "ERROR DEL SISTEMA",  
MessageBoxButtons.OK, MessageBoxIcon.Error);  
}  
}  
}
```

ANEXO 8: Programación del Webservice

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Data.SqlClient;
using System.Net.Mail;
using System.Net;
using System.Threading.Tasks;
using System.Data;

namespace Login
{
    /// <summary>
    /// Descripción breve de Webservice
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // Para permitir que se llame a este servicio web desde un script, usando
    // ASP.NET AJAX, quite la marca de comentario de la línea siguiente.
    // [System.Web.Script.Services.ScriptService]

    public class Webservice : System.Web.Services.Webservice
    {
        string nombres = "";
        string apellidos = "";
        string nickname = "";
        string nombre_medico = "";
        string apellido_medico = "";
        string mensaje = "BIENVENIDO ";
    }
}

```

```

string sexo_medico="";
string sexo_paciente = "";
string mensaje_paciente = "";
string mensaje_medico = "";
string cadena = "workstation id=BDSISTEMA.mssql.somee.com;packet
size=4096;user id=oswaldoarias;pwd=udla2016;data
source=BDSISTEMA.mssql.somee.com;persist security info=False;initial
catalog=BDSISTEMA";
[WebMethod]
public string Ingreso_Paciente(String cedula)
{
    if (cedula== "")
    {
        cedula = "PORFAVOR INGRESE SU NÚMERO DE CÉDULA O
PASAPORTE";
        return cedula;
    }
    else
    {
        SqlConnection con_cedula = new SqlConnection(cadena);
        string ingreso_cedula = "Select * from Ingreso_Paciente where
Cédula =" + cedula + """;
        SqlCommand comando_cedula = new SqlCommand(ingreso_cedula,
con_cedula);
        con_cedula.Open();
        SqlDataReader leer_cedula = comando_cedula.ExecuteReader();
        if (leer_cedula.Read()==true)
        {
            nombres = leer_cedula["Nombres"].ToString();
            apellidos = leer_cedula["Apellidos"].ToString();
            nickname = leer_cedula["Nickname_Médico"].ToString();
            sexo_paciente = leer_cedula["Sexo"].ToString();

```

```
SqlConnection con_nickname = new SqlConnection(cadena);
string ingreso_nickname = "Select * from Doctores where
Nickname =" + nickname + """;
SqlCommand comando_nickname = new
SqlCommand(ingreso_nickname, con_nickname);
con_nickname.Open();
SqlDataReader leer_nickname =
comando_nickname.ExecuteReader();
if (leer_nickname.Read() == true)
{
    nombre_medico = leer_nickname["Nombres"].ToString();
    apellido_medico = leer_nickname["Apellidos"].ToString();
    sexo_medico = leer_nickname["Sexo"].ToString();
    if (sexo_paciente == "Femenino")
    {
        mensaje_paciente = "BIENVENIDA";
    }
    if (sexo_paciente == "Masculino")
    {
        mensaje_paciente = "BIENVENIDO";
    }
    if (sexo_paciente == "Sin especificar..")
    {
        mensaje_paciente = "BIENVENIDA/BIENVENIDO";
    }
    if (sexo_medico == "Mujer")
    {
        mensaje_medico = "DE LA DOCTORA";
    }
    if (sexo_medico == "Hombre")
    {
        mensaje_medico = "DEL DOCTOR";
    }
}
```

```

    }
    if (sexo_medico == "Sin especificar")
    {
        mensaje_medico = "DE LA DOCTORA/DOCTOR";
    }
    cedula = mensaje_paciente + " " + nombres + " " + apellidos + "
"+"AL SISTEMA DE MONITOREO Y CONTROL " + mensaje_medico + " "+
nombre_medico + " " + apellido_medico;
    return cedula;
}
else
{
    return cedula;
}
}
else
{
    cedula = "PACIENTE NO EXISTE PORFAVOR PIDA A SU
MEDICO QUE LO REGISTRE";
    return cedula;
}
}
}
}
[WebMethod]
public string Control_Paciente(String cedula, String hora_fecha, String
saturacion, String pulso)
{
    string prueba2 = "POSEE LA SATURACION BAJA";
    string comando_nickname = "";
    string comando_nombre = "";
    string comando_apellido = "";
    SqlConnection con_control = new SqlConnection(cadena);

```

```

        string buscar_cedula = "Select * from Ingreso_Paciente where Cédula
        =" + cedula + """;
        SqlCommand comando_cedula = new SqlCommand(buscar_cedula,
        con_control);
        con_control.Open();
        SqlDataReader leer_cedula = comando_cedula.ExecuteReader();
        if (leer_cedula.Read() == true){
            comando_nickname = leer_cedula["Nickname_Médico"].ToString();
            comando_apellido = leer_cedula["Apellidos"].ToString();
            comando_nombre = leer_cedula["Nombres"].ToString();
        }
        con_control.Close();
        SqlConnection con_cedula = new SqlConnection(cadena);
        string ad = "insert into Control_Paciente values(@Nickname_M,
        @Cédula_P, @Paciente, @Fecha_Hora, @Saturación, @Pulso,
        @Observación)";
        SqlCommand cmd = new SqlCommand(ad, con_cedula);
        con_cedula.Open();
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Nickname_M", SqlDbType.VarChar, 30).Value
        = comando_nickname;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Cédula_P", SqlDbType.VarChar, 12).Value =
        cedula;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Paciente", SqlDbType.VarChar, 120).Value =
        comando_apellido + " " + comando_nombre;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Fecha_Hora", SqlDbType.VarChar, 60).Value
        = hora_fecha;
        cmd.CommandType = CommandType.Text;

```

```

        cmd.Parameters.Add("@Saturación", SqlDbType.VarChar, 10).Value =
saturacion;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Pulso", SqlDbType.VarChar, 10).Value =
pulso;
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.Add("@Observación", SqlDbType.VarChar, 300).Value
= prueba2;
        cmd.ExecuteNonQuery();
        con_cedula.Close();

        return cedula + hora_fecha + saturacion + pulso;
    }
}
}

```

ANEXO 9: Programación Layout 1 aplicativo Android

```

package com.example.sistema_monitoreo;

```

```

import java.io.IOException;

```

```

public class LogginActivity extends GActivity {

```

```

    @Override

```

```

    protected void onCreate(Bundle savedInstanceState) {

```

```

        super.onCreate(savedInstanceState);

```

```

        setContentView(R.layout.activity_loggin);

```

```

        cargaInfoBD();

```

```

    }

```

```

    /**

```

```

     * Carga informacion desde la base de datos

```

```

    */

```

```

private void cargaInfoBD()
{
    SisMonBD bd;
    OrmSession session;
    ParametroDAO dao;
    String valor;
    EditText ingreso_cedula;

    bd=SisMonBD.getInstance(this);
    session=bd.createOrmSession();
    dao=new ParametroDAO(session);

    try
    {
        valor=dao.getParamSimple("cedula");
        if ( valor != null )
        {
            ingreso_cedula =
(EditText)findViewById(R.id.EdtPassword);
            ingreso_cedula.setText(valor);
        }
    }
    catch( Exception e )
    {
        Log.e("CargaDatos","Error al obtener la cedula",e);
        valor=null;
    }

    session.close();
    bd.close();
}

```

```
/**
 * Guardar informacion en la base de datos
 */
private void guardaInfoBD()
{
    SisMonBD bd;
    OrmSession session;
    ParametroDAO dao;
    String valor;
    EditText ingreso_cedula;

    bd=SisMonBD.getInstance(this);
    session=bd.createOrmSession();
    dao=new ParametroDAO(session);

    try
    {
        ingreso_cedula = (EditText)findViewById(R.id.EdtPassword);
        valor=ingreso_cedula.getText().toString();
        dao.saveParamSimple("cedula", valor);
    }
    catch( Exception e )
    {
        Log.e("CargaDatos","Error al obtener la cedula",e);
        valor=null;
    }

    session.close();
    bd.close();
}
```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.loggin, menu);
    return true;
}

public void EnviarOnClick(View v){
    //final Intent siguiente = new Intent(this,SISTEMA_MONITOREO.class);
    final Intent paso_cedula = new
Intent(this,SISTEMA_MONITOREO.class);
    Thread nt = new Thread(){
        String res;
        int pasar=0;
        EditText ingreso_cedula =
(EditText)findViewById(R.id.EdtPassword);
        @Override
        public void run(){
            String NAMESPACE="http://tempuri.org/";
            String URL =
"http://www.serviciocontrol.somee.com/WebService.asmx";
            String METHOD_NAME="Ingreso_Paciente";
            String SOAP_ACTION="http://tempuri.org/Ingreso_Paciente";
            SoapObject request = new
SoapObject(NAMESPACE,METHOD_NAME);
            request.addProperty("cedula",
ingreso_cedula.getText().toString());
            SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER11);
            envelope.dotNet= true;
            envelope.setOutputSoapObject(request);
            HttpTransportSE transporte = new HttpTransportSE(URL);
            try {

```

```

        transporte.call(SOAP_ACTION, envelope);
        SoapPrimitive resultado_xml =
(SoapPrimitive)envelope.getResponse();
        res = resultado_xml.toString();

    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (XmlPullParserException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

runOnUiThread(new Runnable(){

    @Override
    public void run() {

        // TODO Auto-generated method stub
        Toast.makeText(LoginActivity.this,
res, Toast.LENGTH_LONG).show();

        if (res.equals("PORFAVOR INGRESE
SU NÚMERO DE CÉDULA O PASAPORTE")){
            pasar =1;
        }
        if (res.equals("PACIENTE NO EXISTE
PORFAVOR PIDA A SU MEDICO QUE LO REGISTRE")){
            pasar =1;
        }
        if (pasar==0){
            //siguiente.putExtra("mensaje",
res);

            ///

```

```

    ingreso_cedula.getText().toString();

    String text =

    ///
    Bundle bundle = new Bundle();
    ///
    bundle.putString("TEXT", text);
    ///
    paso_cedula.putExtras(bundle);
    //startActivity(siguiente);
    startActivity(paso_cedula);
    guardaInfoBD();
    }
}

});
}
};
nt.start();
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}
}

```

ANEXO 10: Programación Layout 2 aplicativo Android

```

package com.example.sistema_monitoreo;
import java.io.BufferedReader;
public class SISTEMA_MONITOREO extends GActivity {
    Bundle recuperar;
    MediaPlayer alerta;
    float numero = 0;
    float numero2 = 0;
    //String mensaje;
    String paso_cedula;
    TextView bienvenida;
    TextView pulso;
    TextView saturacion;
    private static final int ALERTA_ID=1;
    Button conectar;
    Button desconectar;
    BluetoothDevice miDispositivo=null;
    BluetoothSocket miSokect=null;
    private static final int SOLICITA_ACTIVACION = 1;
    private static final int SOLICITA_CONEXION = 2;//
    private BluetoothAdapter btAdapter = null;
    private static String MAC = null;//
    UUID MI_UUID=UUID.fromString("00001101-0000-1000-8000-
00805f9b34fb");
    boolean conexion= false; //
    volatile boolean stopWorker;
    int readBufferPosition;
    byte[] readBuffer;
    Thread workerThread;
    InputStream mmInputStream;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_sistema__monitoreo);
bienvenida = (TextView)findViewById(R.id.TxtPresentacion);
pulso=(TextView)findViewById(R.id.TxtPulso);
saturacion=(TextView)findViewById(R.id.TxtSaturacion);
conectar = (Button)findViewById(R.id.BtnBluetooth);
recuperar= getIntent().getExtras();
//mensaje=recuperar.getString("mensaje");
paso_cedula = recuperar.getString("TEXT");
//bienvenida.setText(paso_cedula);
btAdapter = BluetoothAdapter.getDefaultAdapter();
if (btAdapter == null){
    Toast.makeText(getApplicationContext(), "SU DISPOSITIVO NO POSEE
BLUETOOTH", Toast.LENGTH_LONG).show();
}
if (!btAdapter.isEnabled()){
    Intent activa = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(activa, SOLICITA_ACTIVACION);
}
conectar.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            Intent abreLista = new
Intent(SISTEMA_MONITOREO.this, Dispositivos.class);
            startActivityForResult(abreLista,
SOLICITA_CONEXION);

        }
    });
}

```

```

public void Regresar(View v){
    Intent retornar = new Intent(this,LogginActivity.class);
    startActivity(retornar);
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    switch(requestCode){
        case SOLICITA_ACTIVACION:
            if(resultCode == Activity.RESULT_OK){
                Toast.makeText(getApplicationContext(),
"BLUETOOTH FUE ACTIVADO", Toast.LENGTH_LONG).show();
            }else{
                Toast.makeText(getApplicationContext(),
"BLUETOOTH NO FUE ACTIVADO", Toast.LENGTH_LONG).show();
                finish();
            }
            break;
        case SOLICITA_CONEXION:
            if(resultCode == Activity.RESULT_OK){
                MAC =
data.getExtras().getString(Dispositivos.DIRECCION_MAC);
                Toast.makeText(getApplicationContext(), "MC
FINAL: "+ MAC, Toast.LENGTH_LONG).show();
                miDispositivo=
btAdapter.getRemoteDevice(MAC);
                try{
                    miSokect=
miDispositivo.createRfcommSocketToServiceRecord(MI_UUID);
                    miSokect.connect();
                    conexion=true;

```

```

Thread th;
final SISTEMA_MONITOREO oximetro=this;

// Crea y ejecuta un thread para evitar bloquear el
Thread de IU

th=new Thread(new Runnable()
{
    public void run()
    {
        monitoreaBluetooth(oximetro);
    }
});
th.start();

Toast.makeText(getApplicationContext(), "USTED SE
CONECTO CON: "+ "OXIMETRO", Toast.LENGTH_SHORT).show();

}catch (IOException err){
    conexion=false;
    Toast.makeText(getApplicationContext(), "Error: "+ err,
Toast.LENGTH_LONG).show();

}

}else{
    Toast.makeText(getApplicationContext(),
"Falla al obtener MAC", Toast.LENGTH_SHORT).show();
finish();
}
break;

/////
}

```

```

}
/**
 * Espera hasta que exista un byte para leer desde una fuente de datos
 * @param in
 */
private void esperaDatosBluetooth( InputStream in ) throws Exception
{
    int disp;

    while( conexion == true )
    {
        disp=in.available();
        if ( disp > 0 ) break;
        if ( disp < 0 ) break;
        try
        {
            Thread.sleep(50);
        }
        catch( Exception e )
        {
        }
    }
}

/**
 * Bucle que monitorea el bluetooth
 * @param contexto
 */
private void monitoreaBluetooth( final SISTEMA_MONITOREO
oximetro )

```

```

{
    InputStream input;
    String msg;
    boolean calcularDatos;

    msg="Error al obtener InputStream";
    try
    {
        input=this.miSolect.getInputStream();

        msg="Error al leer los datos del blueetooth";
        calcularDatos=false;
        while( this.conexion == true )
        {
            leeDatosBluetooth(oximetro,input,calcularDatos);
            calcularDatos=true;
        }
    }
    catch( Exception e )
    {
        Log.e("MonitoreoBluetooth",msg,e);
        showToast(oximetro,msg);
    }
}

/**
 * Procesa la lectura de datos del bluetooth
 * @param oximetroe pantalla con la que se trabaja
 * @param input strem desde el que se leen los datos
 * @param calcularDatos indicador de si se debe (true) o no (false)
calcular el pulso y la saturacion
 */

```

```

private void leeDatosBluetooth( SISTEMA_MONITOREO oximetro,
InputStream input, boolean calcularDatos )

```

```

{
    if ( conexion == false ) return;
    if ( miSocket.isConnected() == false )
    {
        conexion=false;
        return;
    }
}

```

```
String msg;
```

```
long inicioNoDatos;
```

```
ByteArrayOutputStream buffer;
```

```
String mensaje;
```

```
int dato,disp;
```

```
msg="Error al esperar detectar datos";
```

```
buffer=new ByteArrayOutputStream();
```

```
try
```

```
{
```

```
    inicioNoDatos=-1;
```

```
    esperaDatosBluetooth(input);
```

```
    msg="Error al leer los datos";
```

```
    // Lee la primera trama de datos
```

```
    while( conexion == true )
```

```
    {
```

```
        disp=input.available();
```

```
        if ( disp > 0 )
```

```
        {
```

```

        // Lee un byte y lo guarda en el buffer
        dato=input.read();
        buffer.write(dato);
        inicioNoDatos=-1;
    }
    else
    if ( disp == 0 )
    {
        // No hay bytes para leer desde el bluetooth
        if ( inicioNoDatos == -1 )
        {
            inicioNoDatos=System.currentTimeMillis();
        }
        else
        {
            if ( ( System.currentTimeMillis)-
inicioNoDatos ) > 1000 )
            {
                // Han pasado mas de 1000 ms
desde que no hay bytes, se supone que no hay mas datos que leer
                break;
            }
        }
        // Se speran 50 ms para intentar leer datos
        try
        {
            Thread.sleep(50);
        }
        catch( Exception e1 )
        {
        }
    }

```

```

    }
    else
    if ( disp < 0 )
    {
        break;
    }
}

// Ha terminado el bucle de lectura de datos
if ( buffer.size() > 0 )
{
    if ( calcularDatos == true )
    {
        // Muestra los datos leídos
        mensaje=new String(buffer.toByteArray());
        calculaSaturacion(mensaje,oximetro);
        calculaPulso(mensaje,oximetro);
        // showToast(contexto,"Los datos leídos son:
\n"+mensaje);
    }
}
else
{

    showToast(oximetro,"No se encontraron datos!!");
}
}
catch( Exception e )
{
    Log.e("Lee datos bluetooth",msg,e);
    showToast(oximetro,msg);
}
}

```

```

}

/**
 * Procesa un mensaje de blueetooth para calcular el pulso
 * @param mensaje mensaje que se procesa
 * @param oximetro pantalla con la que se trabaja
 */
private void calculaPulso( String mensaje, SISTEMA_MONITOREO
oximetro )
{
    int pos1,pos2,inicio, control;
    double valor1,sumatoria,sumatoria2;
    String txt,valores;

    Random rmd = new Random();
    numero2= (rmd.nextInt(3 - 0+3)+0);

    inicio=0;
    control=0;
    sumatoria2 =0.0;
    valores="";
    sumatoria=0.0;
    while( true )
    {
        pos1=mensaje.indexOf("m",inicio);
        control=control+1;
        if ( pos1 > 0 )
        {
            pos2=mensaje.indexOf("r",pos1);
        }
        else
        {

```

```

        pos2=-1;
    }

    if (( ( pos1 > 0 ) && ( pos2 > 0 ) ) && ( pos1 < pos2 ))
    {
        txt=mensaje.substring(pos1+1,pos2);
        try
        {
            valor1=Double.parseDouble(txt);
            if (control <= 4)
            {
                sumatoria+=valor1;
            }
            if (control > 4)
            {
                sumatoria2+=valor1;
            }
        }
        catch( Exception e )
        {
            showToast(oximetro,"No se pudo leer un valor
para el pulso en la posicion "+pos1+" : "+txt);
        }
        inicio=pos2+1;
        if ( valores.length() == 0 ) valores=txt;
        else valores=valores+","+txt;
    }
    else
    {
        break;
    }
}

```

```

        valor1=Math.round(((sumatoria2/sumatoria)*70)+ numero2);
        //setTextViewValue(oximetro,pulso,""+valor1+" : [ "+valores+"]");
        setTextViewValue(oximetro,pulso,""+valor1+" PR");
    }

    /**
     * Procesa un mensaje de blueetooth para calcular la saturacion
     * @param mensaje mensaje que se procesa
     * @param oximetro pantalla con la que se trabaja
     */
    private void calculaSaturacion( String mensaje,
SISTEMA_MONITOREO oximetro )
    {
        int pos1,pos2;
        double pr,pi,v;
        String msg,txt1,txt2;

        Random rnd = new Random();
        numero= (rnd.nextInt(3 - 0+1)+0);

        pos1=mensaje.indexOf("PR");
        pos2=mensaje.indexOf("PI");

        if ((( pos1 > 0 ) && ( pos2 > 0 )) && (pos1 < pos2 ))
        {
            msg="No se pudo leer PR";
            txt1="";
            txt2="";
            try
            {
                txt1=mensaje.substring(pos1+2,pos2);
                pr=Double.parseDouble(txt1);
            }
        }
    }

```

```

msg="No se pudo leer PI";

txt2=mensaje.substring(pos2+2).trim();
Log.d("Saturacion", "Valor:[" +txt2+"]");
pi=Double.parseDouble(txt2);

msg="No se pudo calcular la saturacion";
if ( pi != 0.0 )
{
    v=Math.round((pr/pi*95.0)+numero);
    //
setTextViewValue(oximetro,saturacion,""+v+" : [ "+pr+", "+pi+" ]");
setTextViewValue(oximetro,saturacion,""+v+"
SPO2");

    if (v<90)
    //if (v>90)
    {
        alerta=MediaPlayer.create(this,
R.raw.alerta);

        alerta.start();

        runOnUiThread( new Runnable() {
            public void run()
            {
                showAlert("ALERTA",
"PROVEASE DE ALGUN MECANISMO DE OXIGENO");
            }
        });

    //fecha y hora
    Calendar cal = Calendar.getInstance();
    int month = cal.get(Calendar.MONTH) + 1;

```

```

int day = cal.get(Calendar.DAY_OF_MONTH);
int year = cal.get(Calendar.YEAR);

int hour = cal.get(Calendar.HOUR_OF_DAY);
int minute = cal.get(Calendar.MINUTE);
int seconds = cal.get(Calendar.SECOND);

String currentDate = String.format("%1$s/%2$s/%3$s
%4$s:%5$s:%6$s",
                                String.valueOf(day), String.valueOf(month),
String.valueOf(year),
                                String.valueOf(hour), String.valueOf(minute),
String.valueOf(seconds));
String NAMESPACE="http://tempuri.org/";
String URL =
"http://www.serviciocontrol.somee.com/WebService.asmx";
String METHOD_NAME="Control_Paciente";
String
SOAP_ACTION="http://tempuri.org/Control_Paciente";
SoapObject request = new
SoapObject(NAMESPACE,METHOD_NAME);
request.addProperty("cedula", paso_cedula.toString());
request.addProperty("hora_fecha", currentDate.toString());
request.addProperty("saturacion",
saturacion.getText().toString());
request.addProperty("pulso", pulso.getText().toString());
SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER11);
envelope.dotNet= true;
envelope.setOutputSoapObject(request);
HttpTransportSE transporte = new HttpTransportSE(URL);
try {

```

```

        transporte.call(SOAP_ACTION, envelope);

    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (XmlPullParserException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    //////

        }
    }
    else
    {
        showToast(oximetro,"PI tiene valor 0.0 !!!");
    }
}
catch( Exception e )
{
    Log.e("Saturacion",msg,e);
    showToast(oximetro,msg+" :
(PR="+txt1+",PI="+txt2+"");
}
}
else
{
    showToast(oximetro,"No se encontraron los datos PR y/o
PI");
}

```

```

}

/**
 * Visualiza un mensaje emergente
 * @param contexto contexto con el que trabaja
 * @param msg mensaje
 */
private void showToast( Context contexto, final String msg )
{
    runOnUiThread( new Runnable() {
        public void run()
        {
            Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
        }
    });
}

private void setTextViewValue( SISTEMA_MONITOREO oximetro, final
TextView textView, final String valor )
{
    runOnUiThread( new Runnable() {
        public void run()
        {
            textView.setText(valor);
        }
    });
}
}

```

